
École Doctorale Jean-Henri Lambert

Mémoire
d'Habilitation à Diriger des Recherches

Université de Haute Alsace

DOMAINE DE RECHERCHE : Informatique

Métaheuristiques Hybrides et Leurs Applications

Lhassane IDOUMGHAR

Soutenance le 21 novembre 2012

Composition du jury :

M. Jin-Kao HAO	Professeur des Universités	Rapporteur
M. Abderrafiaa KOUKAM	Professeur des Universités	Rapporteur
M. Patrick SIARRY	Professeur des Universités	Rapporteur
Mme Annie CHOQUET	Professeur des Universités	Examinateur
M. Pierre COLLET	Professeur des Universités	Examinateur
M. Mahmoud MELKEMI	Professeur des Universités	Examinateur
M. Abdellatif MIRAOUI	Professeur des Universités	Examinateur
M. René SCHOTT	Professeur des Universités	Examinateur

Remerciements

Je voudrais tout d'abord remercier l'ensemble des rapporteurs qui m'ont fait honneur d'accepter de faire partie de la liste des six rapporteurs que j'ai soumis au Conseil Scientifique (CS) de l'Université de Haute Alsace. Plus particulièrement, j'adresse mes plus respectueux remerciements à Messieurs Patrick SIARRY, Professeur à l'Université de Paris-Est Créteil, Jin-Kao HAO, Professeur à l'Université d'Angers, Abderrafiaa KOUKAM, Professeur à l'Université de Technologie Belfort Montbéliard, désignés par le CS pour rapporter sur ce travail. Qu'ils soient assurés de ma plus profonde reconnaissance pour l'attention qu'ils apportent à ce mémoire et pour le temps qu'ils vont consacrer à son évaluation. Conscient du travail qu'il me reste encore à effectuer et de ce qui est perfectible, je m'attacherai à suivre humblement leurs conseils et recommandations avisés.

Aussi, je remercie Mme Annie CHOQUET-GENIET, Professeur à l'Université de Poitiers, Monsieur Pierre COLLET, Professeur à l'Université de Strasbourg et M. Abdellatif MIRAOUI, Professeur à l'Université de Technologie Belfort Montbéliard et président de l'université de Marrakech (Maroc) pour avoir accepté d'examiner ce travail.

Mes sincères remerciements sont également adressés au Professeur René Schott et au Professeur Mahmoud Melkemi ainsi qu'au Professeur Patrick SIARRY pour avoir contribué à améliorer la qualité et la clarté de ce mémoire.

Mon remerciement collégial est adressé aux différents collègues (LMIA, LORIA et IRTES) pour m'avoir toujours accueilli dans leurs bureaux pour l'échange d'idées.

Avant de terminer, mes sincères remerciements à ma femme qui a su me soutenir avec toute son affection et qui a fait l'effort de me supporter dans les moments les plus difficiles.

Aussi il serait injuste d'exclure de ces remerciements ceux qui ont contribué indirectement à ce travail par leur présence et leur soutien : mes parents et beaux-parents, mes deux frères et ma sœur.

Table des matières

Table des matières

Résumé & Abstract	xxv
Introduction générale	1
1 Historique de mes activités	1
1.1 Thèse de doctorat CIFRE (01/02/2000-12/12/2002)	1
1.2 Poste ATER et Maître de conférences à l'Université de Haute Alsace, Mulhouse	2
2 Contexte et objectifs	3
3 Plan du mémoire	4
I État de l'art sur les métahéuristiques	7
1 Vocabulaire et définitions	7
2 Problème d'optimisation mono-objectif	8
2.1 Définition	8
2.2 Méthodes exactes	9
2.3 Métahéuristiques	11
2.3.1 Recuit simulé	13
2.3.2 Méthode Tabou	15
2.3.3 Algorithmes génétiques	16
2.3.4 Colonie de fourmis	19
2.3.5 Optimisation par essaim particulaire	21

Table des matières

2.4	Métaheuristiques hybrides	23
2.4.1	PSO hybridé avec une recherche locale	24
2.4.2	PSO intégrant un opérateur génétique	24
3	Problème d'optimisation multiobjectif	26
3.1	Introduction	26
3.2	Méthodes de résolution	26
4	Conclusion	30
II	Métaheuristiques pour la planification des réseaux de radiodiffusion	31
1	Introduction	31
2	Modélisation par contraintes	33
2.1	Description	33
2.2	Algorithme génétique hybride	34
2.3	Résultats et simulations	37
2.3.1	Comparaison en terme de contraintes violées	38
2.3.2	Validation métier des plans de fréquences	40
2.4	Discussion	43
3	Modélisation par histogramme	45
3.1	Méthode tabou probabilisée	50
3.2	Validation métier	51
4	Discussion	55
III	Gestion intelligente de l'énergie dans les smart grids	57
1	Introduction	57
2	Architectures de contrôle décentralisées	58
2.1	Architecture d'un système multi-agents	58
2.1.1	Les Systèmes multi-agents	59
2.1.2	Intérêt pour un système de gestion d'énergie	59
2.1.3	Architecture sélectionnée	60
2.1.4	Structure d'un agent	62
2.2	Interactions entre les agents	63
2.2.1	Changement dans l'architecture SMA	63
2.2.2	Fonctionnement normal du système	64
3	Optimisation du fonctionnement des micro-réseaux et des centrales	65
3.1	Algorithmes d'optimisation	65

3.1.1	Algorithme <i>MPSOM</i> (<i>Metropolis Particle Swarm Optimizaion with Mutation operator</i>)	65
3.1.2	Principe de l'algorithme ICA (<i>Imperialist Competitive Algorithm</i>)	68
3.2	Problème du <i>dispatching</i>	71
3.3	Problème de la <i>planification</i>	72
4	Exemples d'applications	72
4.1	Micro-réseau	73
4.2	Centrale de production	74
5	Conclusion	75
IV Réduction de l'énergie consommée en mémoire SPM		77
1	Introduction	77
2	Problématique	78
2.1	Composants de la hiérarchie mémoire	78
2.1.1	Mémoire Cache	79
2.1.2	Mémoire <i>Scratch-Pad</i>	80
2.1.3	Mémoire Principale	81
2.2	Description de l'architecture mémoire considérée	81
2.3	Formalisation du problème	83
3	Programmes tests utilisés	84
4	Algorithmes de résolution	85
4.1	Synthèse des heuristiques classiques utilisées	85
4.2	Métaheuristiques hybrides	86
4.2.1	Algorithme génétique hybride	86
4.2.2	Algorithme d'optimisation par essaim particulaire hybride	88
4.2.3	Comparaison entre les algorithmes hybrides séquentiels	92
4.2.4	Algorithme GASA distribué	93
4.2.5	Algorithme PSOSA distribué	95
5	Discussion	96
V Métaheuristiques pour l'optimisation de moteurs électriques		97
1	Introduction	98
2	Modélisation et résolution du problème	102
2.1	Optimisation mono-objectif	102
2.2	Optimisation multiobjectif	103

3	Discussion	106
VI	Métaheuristiques pour l'optimisation continue	107
1	Introduction	108
2	Description de quelques <i>benchmarks</i> mathématiques	108
2.1	Fonction de Rastrigin	109
2.2	Fonction Sphère	109
2.3	Fonction de Griewank	110
2.4	Fonction de Rosenbrock	111
2.5	Fonction bicarrée bruitée	111
2.6	Fonction de Schwefel	112
2.7	Fonction d'Ackley	113
2.8	Fonction de Michalewicz	113
2.9	Fonction d'Himmelblau	114
2.10	Fonction de Shubert	115
3	Optimisation des <i>benchmarks</i> mathématiques par l'algorithme <i>PSOSA</i>	115
3.1	Comparaison avec l'algorithme <i>TL-PSO</i>	116
3.2	Comparaison avec les algorithmes <i>PSO</i> , <i>ATREPSO</i> , <i>QIPSO</i> et <i>GMPSO</i>	116
3.3	Tests statistiques	118
3.3.1	Test de Friedman	119
4	Optimisation des <i>benchmarks</i> mathématiques par l'algorithme <i>MPSOM</i>	122
4.1	Comparaison avec les algorithmes <i>GPSO-J</i> , <i>CLPSO</i> et <i>ATM-PSO</i>	122
4.2	Taux de convergence $Q_{measure}$	124
4.3	Test de Friedman	125
5	Conclusion	127
Conclusion et projet scientifique		129
Références bibliographiques		139
Annexe : Quelques unes de mes publications		157

Curriculum VitÆ

Curriculum Vitæ

Curriculum vitae

Diplômes

- 2000-2003 **Doctorat en Informatique**, Période : 02/2000-12/12/2002, Titre de la thèse : *Méthodes algorithmiques pour l'allocation de fréquences*, Thèse Cifre-TDF préparée au Loria et soutenue le 12 décembre 2002 à l'Université Henri Poincaré Nancy 1.
- 1998-1999 **DEA en Informatique**, Université Picardie Jules Verne, Mémoire intitulé : *Proposition d'un modèle de gestion de Qualité de Service dans l'Architecture Corba*.
- 1997-1998 **Maîtrise en Informatique**, Université Ibnou Zohr, Agadir, Maroc.
- 1995-1996 **DEUG Physique-chimie**, Université Ibnou Zohr, Agadir, Maroc.

Fonctions exercées

- Depuis 2004 **Maître de Conférences en Informatique**, Université de Haute Alsace.
- Sept. 2003 – Août 2004 **Attaché Temporaire d'Enseignement et de Recherches (ATER) en Informatique**, Université de Haute Alsace.
- Jan. 2003 – Juin 2003 **Agent temporaire vacataire d'enseignement en Informatique**, Université Henri Poincaré.
- Fév. 2000 – Jan. 2003 **Chargé de Recherche & Développement en Informatique**, Centre d'étude en Radiodiffusion et Radiocommunication de TéléDiffusion de France (TDF-C2R), Metz.
- Sept. 1999 – Jan. 2000 **Agent temporaire vacataire d'enseignement en Informatique**, Institut Universitaire de Technologie, Amiens.
- Fév. 1998 – Août 1998 **Programmeur & Infographiste**, Société À La Carte Publicité, Agadir, Maroc.
- Juil. 1997 – Août 1997 **Stage de formation**, Société OLI_SUD, Agadir, Maroc.

Thèmes et projets de recherche développés

Depuis 2000 **Métaheuristiques pour la planification des réseaux de radiodiffusion, Partenaire : LMIA-UHA, TéléDiffusion de France (TDF) et LORIA-INRIA Grand Est.**

Description : Le secteur des services de communication par voie hertzienne connaît une croissance très forte, tant dans le domaine des radiocommunications que dans le domaine de la radiodiffusion avec l'émergence de nouveaux réseaux de radio ou de télévision numérique. L'ingénierie de tels réseaux fait appel à des techniques de simulation informatique pour le processus dit de planification, qui consistent notamment à évaluer la couverture radio du réseau et allouer une portion de spectre hertzien disponible à chacun des émetteurs. La génération d'un plan de fréquences est une tâche difficile dans le processus de planification, qui peut conduire à des plans de fréquences peu adéquats d'un point de vue métier. Dans ce travail, nous avons proposé de nouveaux algorithmes qui ont permis de résoudre efficacement le problème d'allocation de fréquences dans le cadre de la radiodiffusion. Ces algorithmes sont basés essentiellement sur des métaheuristiques et des algorithmes stochastiques hybrides (algorithmes génétiques, processus décisionnels de Markov). L'utilisation de ces nouvelles approches a montré que les solutions obtenues sont nettement meilleures que les meilleures solutions opérationnelles existant dans ce domaine. A l'issue de ce travail le logiciel d'allocation de fréquences ALOES (ALgorithme d'Optimisation de l'Emploi de Spectre), intégrant les différentes méthodes proposées, a été développé. Ce logiciel est actuellement utilisé pour le déploiement en France et en Europe de la Télévision Numérique Terrestre (TNT). Une thèse Cifre est lancée depuis Janvier 2012 et a pour but la mise en place de nouvelles méthodes d'optimisation multiobjectif pour la planification des réseaux de radiodiffusion. L'objectif de cette thèse est de faire coopérer plusieurs types de réseaux de radiodiffusion dans la même bande de fréquences tout en respectant les différentes contraintes liées à chaque type de réseau.

2009 – 2011 **Métaheuristiques pour la réduction de l'énergie consommée en mémoire dans les systèmes embarqués, Partenaire : LMIA-UHA et INRIA-Grand Est.**

Description : Le but de ce travail est de s'attaquer au problème de l'énergie dans les systèmes embarqués temps réel d'un point de vue logiciel. Bien que le problème de l'énergie ait été souvent considéré d'un point de vue matériel, il est bien moins exploré du point de vue logiciel, où les optimisations et les stratégies de gestion mémoire se sont plutôt focalisées sur la vitesse ou la taille. Pourtant, des opportunités considérables existent à ce niveau, car les optimisations énergétiques peuvent souvent être meilleures lorsqu'elles sont faites au niveau logiciel (compilateur, gestion mémoire, ordonnanceur...) plutôt qu'au niveau matériel. En effet, dans le premier cas, la logique de l'optimisation est faite en partie lors de la compilation, sans aucun surcoût à l'exécution, alors que dans le matériel cette logique correspond à des circuits qui utilisent de l'énergie. Ceci permet de dédier beaucoup plus de ressources (temps CPU, mémoire) à la compilation d'un système, ce qui est impossible dans le matériel. Un contexte plus large et plus complet peut donc être pris en compte. Ce qui permet au compilateur de "connaître le futur" de l'exécution d'un programme ou ensemble de programmes, tout particulièrement en ce qui concerne son comportement mémoire. Bien entendu, les meilleurs résultats sont obtenus en combinant optimisations logicielles et matérielles, ou, au moins, en prenant en compte le matériel dans les optimisations logicielles. Le travail que nous avons mené dans ce domaine, et qui fait partie de la thèse de Mme Maha Idrissi-Aouad que j'ai encadrée, nous a permis de mettre en place plusieurs algorithmes hybrides et d'avoir des gains, en termes d'économie d'énergie, très intéressants en comparaison avec les méthodes classiques utilisées dans ce domaine.

2010 – 2011 **Algorithmes hybrides pour la gestion intelligente de l'énergie dans les smart grids : Projet COMCO (COnparaison des Moyens de COntrôle), Partenaire : Institut de Recherche sur les Transports, l'Energie et la Société (IRTES) à Belfort, General Electric à Belfort.**

Description : Le fonctionnement d'une centrale de production d'énergie est basé sur l'utilisation de turbines permettant la combustion de gaz ou de fioul. Ces turbines consommant de grandes quantités de combustible et émettant du gaz NOx (oxyde d'azote), dont les émissions sont contrôlées et facturées, il est important pour les exploitants d'optimiser les coûts de fonctionnement de leur installation, tout en répondant au mieux aux variations de la demande. Une centrale thermique est généralement composée de plusieurs turbines alimentées en carburant (gaz ou fioul), dont les points de fonctionnement sont déterminés en fonction de la charge totale mesurée (c'est-à-dire la demande totale de puissance), qu'il s'agisse d'habitations ou d'une usine. Les méthodes de gestion actuelles permettent de gérer plusieurs turbines en même temps, mais pas d'intégrer d'autres composants tels que du stockage (par batteries par exemple) ou une connexion au réseau pour revendre le surplus éventuel d'énergie en fonction de son coût de rachat. Une nouvelle méthode doit donc être proposée pour pallier ce manque, et gérer l'ensemble en garantissant à la fois un coût minimal et la stabilité du réseau. Dans ce projet, nous avons proposé un algorithme basé sur un système multi-agents. En permettant, de par sa structure, la connexion et la déconnexion à chaud de matériel (turbines, stockage ou autre), ainsi que la gestion de leur mode de fonctionnement afin de coller au mieux à la demande, le système multi-agents comporte un agent par turbine, unité de stockage et connexion réseau, ainsi qu'un agent de contrôle. Chaque agent permet ensuite de récupérer les informations sur l'état du matériel auquel il est connecté ainsi que de le commander. Un agent central vient ensuite récupérer l'ensemble des données sur le réseau et lance un algorithme d'optimisation, basé sur des métaheuristiques hybrides, permettant de trouver les points optimaux de fonctionnement des éléments, en fonction de leur coût. Les résultats obtenus font partie de la thèse de M. Robin Roche que j'ai co-encadrée.

Depuis 2008 **Métaheuristiques pour l'optimisation de moteurs électriques à aimant permanent, Partnaire : Daniel Fodorean, Professeur Assistant à l'Université de Cluj-Napoca, Roumanie.**

Description : L'objectif principal de ces travaux est de réduire la masse et les pertes d'un "moteur synchrone à aimant permanent" qui constitue l'un des éléments clés d'un véhicule électrique. Ce qui permettra non seulement de minimiser le coût de fabrication et la consommation, mais aussi d'augmenter considérablement l'autonomie et la fiabilité. Il s'agit ici d'un problème d'optimisation multiobjectif qui consiste à maximiser le couple moteur et minimiser sa masse tout en respectant des contraintes de performance à savoir la puissance de sortie, la consommation électrique, le rendement et le facteur de puissance des différents moteurs mis en place. Pour résoudre un tel problème, nous avons mis en place de nouveaux algorithmes d'optimisation basés sur les métaheuristiques hybrides. Les premiers résultats obtenus nous ont permis d'avoir un gain de 47% sur la masse pour de très bonnes performances. Forts de ces résultats, nous avons pu, avec Daniel Fodorean, décrocher deux projets :

- le projet de recherche intitulé "Intelligent Hybrid Vehicle for Individual Transportation of People with Reduced Mobility". Projet financé par le Conseil national de la recherche dans l'enseignement supérieur (acronyme en CNCSIS Roumaine) et qui sera mis en œuvre par l'Université Technique de Cluj-Napoca (UTCN), Cluj, Roumanie. Le projet a débuté le 28 Juillet 2010 et se déroule sur trois années.
- le projet de recherche intitulé "Optimal Energy Management for Light Electric Vehicles". Ce projet (faisant partie d'un groupe de projets financés par la Communauté Européenne, intitulé 4D-POSTDOC) a débuté le 1^{er} Juin 2010 et se déroulera jusqu'au 31 Avril 2013.

Le point commun entre ces deux projets est la conception et l'optimisation de nouvelles structures d'un "moteur synchrone à aimant permanent" pour des scooters électriques à destination des personnes à mobilité réduite.

Depuis 2007	Détection d'une forme vide dans un nuage de points, Partenaire : Equipe MAGE - LMIA Mulhouse. Description : Un grand nombre de problèmes en géométrie algorithmique consistent à placer une forme donnée (généralement cercle, ellipse, carré, rectangle, polygone) dans un ensemble de points tout, en respectant un certain nombre de contraintes. Dans ce travail nous nous intéressons à la résolution du problème de placement, en proposant une nouvelle représentation de forme très générique qui peut représenter des formes non-convexes et des formes avec des trous, en utilisant des concepts purement géométriques. L'algorithme développé a pour avantage de trouver non pas une, mais toutes les solutions existantes, et de fournir une représentation mathématique pratique et compacte de toutes ces solutions. Les résultats obtenus pourraient être facilement étendus à des problèmes impliquant d'autres transformations géométriques (translation, changement d'échelle, etc.). Ils montrent un lien intéressant avec la notion de propriété du disque vide et, par conséquent, les cellules de Voronoï bien connues. Des applications telles que la reconnaissance des formes géométriques ou d'optimisation sont envisageables.
Juin 2012	Projet PHC BRANCUSI 2013, Partnaire : LMIA-UHA et l'Université de Cluj-Napoca, Roumanie. Description : Le projet transversal que nous proposons, en collaboration avec Daniel Fodorean de l'université de Cluj-Napoca pour la partie électrique, tentera d'évaluer l'utilisation de machines simples ou doubles excités pour la production d'énergie éolienne. Plusieurs topologies d'un générateur synchrone à aimant permanent seront évaluées. La contrôlabilité de larges plages de vitesses pour de telles machines sera optimisée en utilisant de nouveaux algorithmes d'optimisation basés sur des métahéuristiques hybrides. L'objectif de ces approches sera de maximiser à la fois la puissance délivrée, le facteur de puissance et le rendement d'une telle unité de production. Un autre but de ce projet sera l'utilisation d'une unité de stockage hybride, basée sur les super-condensateurs vu leur durée de vie qui est plus importante que pour les piles. Nous souhaitons également exploiter les limites des dispositifs de fonctionnement pour réduire le volume/poids du système tout en réduisant les coûts et en maximisant la gestion de l'énergie entre les dispositifs du générateur de puissance pour de faibles vents.
Sept. 2010 – Sept. 2012	Projet ESDI, Partenaires : MIPS-UHA, LMIA-UHA et la société ESDI. Description : Je suis impliqué, depuis septembre 2010, dans le contrat d'accompagnement qui engage le laboratoire MIPS, le laboratoire LMIA et l'entreprise ESDI. En effet la société ESDI souhaite la réalisation d'une étude, d'une part, sur la modélisation d'un environnement d'hébergement d'applications informatiques en ligne et, d'autre part, sur l'aide au diagnostic en cas de défaillance de cet environnement. Dans le cadre de ce projet, je participe aux réunions régulières de suivi avec l'entreprise, à l'accompagnement scientifique et à la rédaction des rapports intermédiaires. Actuellement, nous travaillons sur le développement d'un outil de prédiction des pannes en utilisant les réseaux de neurones.
Mai 2012	Projet BQR 2012 : obtention d'une subvention de 10000 € du Conseil Scientifique de l'Université de Haute Alsace à Mulhouse. Description : Cette demande de subvention est destinée à monter un premier cluster GPU à l'Université de Haute Alsace. Par le biais de l'enseignement, j'ai pu avoir un accord pour renouveler une salle 32 machines. La demande de BQR 2012, classée 1 ^{ère} , va me permettre d'acquérir des cartes GPU de la gamme GTX680. Le cluster ainsi formé va nous permettre de développer des algorithmes évolutionnaires massivement parallèles. La mise en place de ce cluster est prévue pour le mois de novembre 2012.

Mars 2011 – Décembre 2011	Montage d'une bourse de thèse CIFRE, Partenaires : Laboratoire LMIA, Laboratoire LORIA et la société TDF à Metz. Description : Le problème d'allocation de fréquences est très étudié ces dernières années et des résultats existent déjà dans la littérature. L'originalité du travail proposé réside dans la mise en place d'algorithmes d'optimisation multiobjectif pour la planification des réseaux de radiodiffusion et de radiocommunication afin qu'ils puissent cohabiter dans la même bande de fréquences (les différents travaux existants font de la planification dans des bandes de fréquences dédiées). La cohabitation, dans la même bande de fréquences, des réseaux de nature différente va ajouter une nouvelle complexité au problème d'allocation de fréquences et nécessitera la mise en place de nouveaux algorithmes d'optimisation multiobjectif ; d'où le sujet de la thèse de M. Akram Bedoui qui a démarré le 01/01/2012.
2010	Projet BQR 2010 : Demande d'une subvention de 10000 € au Conseil Scientifique de l'Université de Haute Alsace à Mulhouse, Partenaires : Laboratoire LMIA, Laboratoire Set-UTBM. Description : Cette demande de subvention était destinée à compléter l'achat d'une partie du matériel et de financer un stagiaire pour aider à l'implémentation d'une métaheuristique distribuée pour une gestion optimale de la production électrique. Cette demande a été classée 6 ^{ème} sur 20 projets proposés à l'UHA.
2010	Equipex 2010, Partenaires : Plusieurs Laboratoires de l'UHA. Description : J'ai participé au montage du projet NOMAD (NOuvelles Mobilités Acceptables et Durables) en réponse à l'appel à projet Equipex 2010 de l'UHA. J'étais le correspondant pour le laboratoire LMIA. Le projet de l'UHA s'inscrivait essentiellement dans l'axe prioritaire de "l'urgence environnementale et les écotechnologies" précisé dans la Stratégie Nationale de Recherche et d'Innovation. Il contribue à atteindre les objectifs du développement durable dans des conditions acceptables pour les citoyens, et a pour ambition de structurer l'Université de Haute Alsace autour du thème de "la promotion des villes et des mobilités durables". Ce projet était pluridisciplinaire et a mis en lumière les nombreuses synergies apportées par le croisement des idées et des expertises des sciences "dures" et "complexes". Ainsi, les axes de recherches et demandes d'équipements qui s'y rapportent ont pour thème les "Matériaux, et processus pour de NOuvelles Mobilités Acceptables et Durables" NOMAD.

Encadrements

Encadrement des thèses

Depuis Jan. 2012	Encadrement à 50% de la thèse, Cifre avec TDF, de M. Akram Bedoui, Université de Lorraine, Co-encadrement avec le Prof. René Schott (50%), Sujet : Méthodes d'optimisation multiobjectif pour la planification des réseaux de radiodiffusion. Etat d'avancement : état de l'art et modélisation du problème.
Depuis Sept. 2011	Encadrement à 25% de la thèse de Nicolas Chérin, Université de Haute Alsace, Co-encadrement avec le Prof. Mahmoud Melkemi (25%) et Frédéric Corlier MCF (50%), Sujet : Optimisation de la reconstruction de formes 3D. Etat d'avancement : état de l'art et modélisation du problème.
2009 – 2012	Encadrement à 33% de la thèse de M. Robin Roche, Université de Technologie Belfort-Montbéliard, Co-encadrement avec le Prof. Abdellatif Miraoui (33%) et Benjamin BLUNIER MCF-HDR (34%) (Décédé le 23 Février 2012), Sujet : Algorithmes hybrides pour la gestion intelligente de l'énergie dans les smart grids. Etat d'avancement : soutenance le 07 décembre 2012.

- 2008 – 2012 **Encadrement à 50% de la thèse de M. Thomas Iwaszko**, *Université de Haute Alsace*, Co-encadrement avec le Prof. Mahmoud Melkemi (50%), Sujet : Généralisation du diagramme de Voronoï et étude du placement géométrique à base de forme-modèle.
Etat d'avancement : soutenance le 22 novembre 2012.
- 2008 – 2011 **Encadrement à 30% de la thèse de Mme Maha Idrissi-Aouad**, *Université Henri Poincaré Nancy 1*, Co-encadrement avec le Prof. René Schott (34%) et Olivier Zendra CR (33%), Sujet : Conception d'algorithmes hybrides pour l'optimisation de l'énergie mémoire dans les systèmes embarqués et de fonctions multimodales.
Thèse soutenue le 4 juillet 2011.

Encadrement des stages de Master recherche

- Jan. 2012 – Août 2013 **Encadrement à 33% de stage de M. Renaud Vanhoutreve**, *Master 1^{ère} et 2^{ème} année Informatique - Ingénierie des logiciels et des connaissances : Apprentissage*, Co-encadrement avec Jean-Louis Paillaud, IS2M et le Prof. Pierre-Collet LSiiT-UDS, Sujet : Algorithmes évolutionnaires sur GPU pour la résolution structurale des zéolithes.
La 1^{ère} partie du stage est soutenue le 28 août 2012.
- Mars – Août 2012 **Encadrement à 50% de stage de Mme Imene Ben-Yahia**, *Master 2^{ème} année Automatique & Informatique Industrielle, spécialité : Systèmes embarqués et Communicants*, Co-encadrement avec Abdelhafid Abouaissa, GRTC-MIPS-UHA, Sujet : Développement d'un algorithme d'optimisation multiobjectif pour la gestion d'énergie dans un réseau de capteurs.
Soutenance de stage prévue le 05 septembre 2012.
- Fév. – Juin 2012 **Encadrement à 33% d'un stage d'initiation à la recherche**, *Ecole IGA, Casablanca, Maroc*, Co-encadrement avec Mme Maha Idrissi-Aouad, Sujet : Algorithme génétique hybride pour la gestion de l'emploi du temps.
Stage soutenu en juin 2012.
- Fév. – Août 2011 **Encadrement à 100% de stage de M. Nicolas Chérin**, *Master Informatique spécialité : Algorithmique, Hautes Performances et Modélisation (AHPM)*, Sujet : Métaheuristiques hybrides pour l'optimisation multiobjectif.
Stage soutenu en septembre 2011.
- Fév. – Mai 2010 **Encadrement à 100% d'un stage (Master 1) d'initiation la recherche**, *Université de Haute Alsace*, Sujet : Développement d'une interface graphique 3D pour l'étude de comportement des métahéuristiques.
Stage soutenu en juin 2010.

Encadrement des stages pour les licences et Masters professionnelles

- Mai – Juin 2000 **Encadrement d'un stage d'initiation à la recherche**, *Ecole d'ingénieur ESIAL, Université Henri Poincaré*, Sujet : Proposition d'un algorithme d'allocation de fréquences.
Stage soutenu en juin 2000.
- Mai – Juillet 2007 **Encadrement de deux stages de licence 3 Informatique**, *Université de Haute Alsace*, Sujet : Mise en place des outils graphiques pour la construction des squelettes et la visualisation de la détection de forme.
Stage soutenu en septembre 2007.
- Fév. 2005 – Août 2012 **Encadrement de 16 stages de licence 3^{ème} année et Master 2^{ème} année Miage**, *Université de Haute Alsace*.

32 rue de Ruelisheim, Illzach – 68110

+33 6 99 90 70 33 • +33 3 89 33 60 25 • +33 3 89 33 64 91

• lhassane.idoumghar@uha.fr • www.lmia.uha.fr/~mage/idoumghar/

Juin – Août	Encadrement de stage de fin de deuxième année , <i>SupElec à Metz.</i>
2002	Stage soutenu en Septembre 2002
Avril – Août	Encadrement de stage de fin d'année , <i>DESS ISIAL.</i>
2002	Stage soutenu en septembre 2002
Juin – Août	Encadrement de stage de fin de deuxième année , <i>ESIAL.</i>
2001	Stage soutenu en septembre 2001

Publications

Le tableau suivant résume mes principales publications :

Catégories	Nombre
Revues internationales avec comité de lecture	9
Conférences Internationales avec comité de lecture et actes	21
Conférences nationales avec comité de lecture et actes	3
Brevets	2
Articles soumis	5
Articles de vulgarisation	5
Logiciels réalisés	2

Revues internationales avec comité de lecture

[1] **L. Idoumghar, D. Fodorean, A. N'diaye, D. Bouquain and A. Miraoui**, “*Simulated Annealing Algorithm for the Optimization of an Electrical Machine*”, Journal of IET Electric Power Applications (Accepted for publication, Impact Factor 1.173, in press), juillet 2011.

[2] **R. Roche, L. Idoumghar, S. Suryanarayanan, M. Daggag, C.-A. Solacolu and A. Miraoui**, “*A Flexible and Efficient Multi-Agent Gas Power Plant Energy Management System With Economic and Environmental Constraints*”, Journal of Applied Energy, (Accepted for publication, Impact Factor 5.106, in press), juin 2012.

[3] **D. Fodorean, L. Idoumghar and L. Szabo**, “*Motorization for electric scooter by using permanent magnet machines optimized based on hybrid metaheuristic algorithm*”, Journal of IEEE Transaction on Vehicular Technology, (Accepted for publication, Impact Factor 1.921, in press), juillet 2012.

[4] **L. Idoumghar, M. Idrissi-Aouad, M. Melkemi and R. Schott**, “*Hybrid PSO-SA Type Algorithms for Multi-Modal Function Optimization and Reducing Energy Consumption in Embedded Systems*”, Journal of Applied Computational Intelligence and Soft Computing, Vol. 2011, pp. 1-12 , juin 2011.

[5] **L. Idoumghar and R. Schott**, “*Two Distributed Algorithms for the Frequency Assignment Problem in the Field of RadioBroadcasting*”. Journal of IEEE Transactions on Broadcasting, Vol. 55, N. 2, Part 1, pp. 223-229, June 2009, Impact Factor 1.703.

[6] **L. Idoumghar, T. Raminosoa and A. Miraoui**, “*New Tabu Search Algorithm to Design an Electric Motor*”. Journal of IEEE Transactions on Magnetic, Vol. 45, N. 3, pp. 1498 - 1501, March 2009, Impact Factor 1.363.

[7] **L. Idoumghar, R. Schott and J.Y. Greff**, “*Application of Markov Decision Processes to the Frequency Assignment Problem*”. Journal of Applied Artificial Intelligence, pp. 761-773, Vol. 18, N. 8, September 2004, Impact Factor 0.849.

32 rue de Ruelisheim, Illzach – 68110

+33 6 99 90 70 33 • +33 3 89 33 60 25 • +33 3 89 33 64 91
• lhassane.idoumghar@uha.fr • www.lmia.uha.fr/~mage/idoumghar/

[8] **L. Idoumghar and Ph. Debreux**, "New Modeling Approach for the Frequency Assignment Problem in Broadcasting". Journal of IEEE Transactions on Broadcasting, pp. 293-298, Vol. 48, N. 4, December 2002, Impact Factor 1.703.

[9] **L. Idoumghar, M. Alabau and R. Schott**, "New Hybrid Genetic Algorithms for The Frequency Assignment Problem". Journal of IEEE Transactions on Broadcasting, Vol. 48, N. 1, 27-34, March 2002, Impact Factor 1.703.

Brevets

[1] **R. Roche, L. Idoumghar, A. Miraoui, B. Blunier, M.Daggag, C. Solacolu and A. Boyette**, "Système et procédé de commande d'une installation de production d'énergie électrique", Brevet d'Invention B11-0627FR, General Electric, 2011.

[2] **L. Idoumghar**, "Procédé d'allocation de fréquences à un ensemble d'émetteurs / récepteurs en réseau par un processus génétique hybride", Brevet d'Invention N. 01 06885, TDF-C2R, 25 Mai 2001.

Conférences internationales avec comité de lecture et actes

[1] **R. Roche, L. Idoumghar, B. Blunier and A. Miraoui**, "Imperialist Competitive Algorithm for Dynamic Optimization of Economic Dispatch in Power Systems". Proceedings of International Conference on Artificial Evolution (EA-2011), pp. 1-12, October 24-26, 2011, Angers, France (CD-ROM).

[2] **R. Roche, L. Idoumghar, B. Blunier and A. Miraoui**, "Optimized Fuel Cell Array Energy Management Using Multi-Agent Systems". Proceedings of 46th Annual Conference of IEEE Industry Applications Society (IAS 2011), pp. 1-12, October 9-13, 2011, Orlando, FL, USA.

[3] **L. Idoumghar, M. Idrissi-Aouad and R. Schott**, "Sequential and Cooperative Distributed SA-Type Algorithms for Energy Optimization in Embedded Systems". Proceedings of the 2010 International Conference on Computational Intelligence and Software Engineering (CiSE 2010), pp. 1-4, December 10-12, 2010 in Wuhan, China.

[4] **L. Idoumghar, M. Idrissi-Aouad, M. Melkemi and R. Schott**, "Metropolis Particle Swarm Optimization Algorithm with Mutation Operator For Global Optimization Problems". Proceedings of 22th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2010), pp. 35-42, October 27-29, 2010, Arras, France.

[5] **L. Idoumghar, M. Idrissi-Aouad, R. Schott and O. Zendra**, "Sequential and Distributed Hybrid GA-SA Algorithms for Energy Optimization in Embedded Systems". Proceedings of IADIS International Conference Applied Computing 2010, pp. 167-174, October 14-16, 2010, Timisoara, Romania.

[6] **L. Idoumghar, M. Idrissi-Aouad, R. Schott and O. Zendra**, "Reduction of Energy Consumption in Embedded Systems : A Hybrid Evolutionary Algorithm". Proceedings of the 3rd International Conference on Metaheuristics and Nature Inspired Computing (META 2010), volume 95, October 27-31, 2010, Djerba, Tunisia.

[7] **T. Iwaszko, M. Melkemi and L. Idoumghar**, "A Geometric Data Structure Suitable for Fast Image Mining or Retrieval". Proceedings of International Conference on Image Analysis and

32 rue de Ruelisheim, Illzach – 68110

+33 6 99 90 70 33 • +33 3 89 33 60 25 • +33 3 89 33 64 91

• lhassane.idoumghar@uha.fr • www.lmia.uha.fr/~mage/idoumghar/

Recognition (ICIAR 2010), pp. 364-373, Povoa de Varzim, Portugal, June 21-23, 2010.

[8] **L. Idoumghar, D. Fodorean and A. Miraoui**, "Using Hybrid Constricted Particles Swarm and Simulated Annealing Algorithm for Electric Motor Design". Proceedings of the 14th Biennial IEEE Conference on Electromagnetic Field Computation (CEFC 2010), Chicago - Illinois USA, May 9 - 12, 2010.

[9] **L. Idoumghar, D. Fodorean and A. Miraoui**, "Simulated Annealing Algorithm for multi-objective optimization : Application to Electric Motor Design". Proceedings of the 29th IASTED International Conference on Modelling, Identification and Control (MIC 2010), pp. 190-196, February 15 - 17, 2010.

[10] **L. Idoumghar, L. Cucu and R. Schott**, "Tabu Search Algorithms for the Multiprocessor Scheduling Problem". Proceedings of the 10th IASTED International Conference on Artificial Intelligence and Applications (AIA 2010), pp. 1-5, February 15 - 17, 2010.

[11] **T. Iwaszko, M. Melkemi and L. Idoumghar**, "Regions of Empty Overlapping Circles". Proceedings of the 7th Japan Conference on Computational Geometry and Graphs. November 11 - 13, 2009, Kanazaw, Japan.

[12] **T. Iwaszko, M. Melkemi and L. Idoumghar**, "A theoretical structure for computational geometry : regions of point-free overlapping circles". Proceedings of the 9th WSEAS international conference on Signal processing, computational geometry and artificial vision. pp. 117- 122, 2009.

[13] **L. Idoumghar, M. Melkemi and R. Schott**, "A Novel Hybrid Evolutionary Algorithm for Multi-modal Function Optimization and Engineering Applications". Proceedings of the 13th IASTED, International Conference on Artificial Intelligence and Soft Computing (ASC 2009), pp. 87-93, Palma de Mallorca, Spain, September 7 - 9, 2009.

[14] **L. Idoumghar, T. Raminosoa and A. Miraoui**, "New Tabu Search Algorithm to Design an Electric Motor". Proceedings of the 13th Biennial IEEE Conference on Electromagnetic Field Computation (CEFC 2008). Athens, Greece, May 11 - 15, 2008, CD-ROM.

[15] **L. Idoumghar and M. Melkemi**, "Pattern Retrieval from a Cloud of points Using Geometric Concepts". Proceedings of the 4th International Conference on Image And Recognition (ICIAR 2007), LNCS 4633, pp. 460 - 468. M. Kamel and A. Campilho Eds, Montreal Canada, August 22 - 24, 2007.

[16] **L. Idoumghar and M. Melkemi**, "Voronoi diagram based on a non-convex pattern : an application to extract patterns from a cloud of points". Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007), pp. 282 - 287, July 9 - 12, 2007.

[17] **L. Idoumghar and R. Schott**, "A New Hybrid GA-MDP Algorithm For The Frequency Assignment Problem". Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 06), pp. 18 - 27, IEEE Pub, Washington D.C, USA. November 13 - 15, 2006.

[18] **L. Idoumghar and R. Schott**, "Probabilistic and Algorithmic Methods for the Frequency Assignment Problem". Proceedings of the 5th International Congress on Industrial and Applied Mathematics (ICIAM 2003), Sydney, Australia, July 7 - 11, 2003, p. 228 - 229, Printed by UTS, 2003.

[19] **L. Idoumghar and Ph. Debreux**, "New Modelling Approach for the Frequency Assignment Problem in Broadcasting". Proceedings of the IASTED International Conference on Software Engineering and Applications (SEA 2002), pp. 805 - 812, November 4 - 6, 2002, Combridge USA, M.H. Hamza Ed.

[20] **L. Idoumghar, R. Schott and J.Y Greff**, "Using Markov Decision Processes to Solve the Frequency Assignment Problem". Proceedings of the 6th IASTED International Conference on Artificial intelligence and Soft Computing (ASC 2002), pp. 552 - 557, July 17-19 2002, Banff, Canada, Acta Press Pub.

[21] **L. Idoumghar, M. Alabau and R. Schott**, "New Hybrid Genetic Algorithms for The Frequency Assignment Problem". Proceedings of the 13th International IEEE Conference on Tool with Artificial Intelligence (ICTAI 2001), pp. 136 - 142, November 7 - 9, 2001, Dallas - Texas, USA, IEEE Pub.

Conférences nationales avec comité de lecture et actes

[1] **L. Idoumghar, R. Roche, B. Blunier and A. Miraoui**, "Algorithmes hybrides pour la gestion intelligente de l'énergie dans les smart grids". Proceedings of "7èmes Journées Francophones Planification, Décision, et Apprentissage pour la conduite de systèmes" (JFPDA 2012), pp. 1-12, 22-23 Mai 2012, Nancy.

[2] **L. Idoumghar, M. Alabau and R. Schott**, "Nouvelle Approche Hybride pour Résoudre le Problème d'Allocation de Fréquences". Proceedings of "Reconnaissance des Formes et Intelligence Artificielle" (RFIA 2002), pp. 241-248, 8-10 Janvier 2002, Angers, France, AFRIF-AFIA Pub.

[3] **L. Idoumghar, M. Alabau and R. Schott**, "A New Hybrid Genetic Algorithm for the Graph Colouring Problem". Proceedings of the 3rd Colloquium on Computational Telecommunications (ALGOTEL 2001), pp. 111-117, 28-30 Mai 2001, Saint Jean de Luz, France, INRIA Lorraine Pub.

Travaux en soumission

[1] **R. Roche, L. Idoumghar and A. Miraoui**, "Agent-Based Optimal Fuel Cell Array Energy Management System", Journal of IEEE Transactions on Industry Applications, (Accepted with corrections), juin 2012.

[2] **R. Roche, L. Idoumghar, B. Blunier and A. Miraoui**, "Imperialist Competitive Algorithm for Dynamic Optimization of Economic Dispatch in Power Systems", (Accepted for publication in to the Lecture Notes in Computer Science, Springer September 2012).

[3] **L. Idoumghar**, "Hybrid Algorithms for Multi-Modal Functions Optimization", (submitted to Journal of Mathematics and Computer Science).

[4] **M. Melkemi, L. Idoumghar and T. Iwaszco**, "Shape Placement in a Set of Points : Finding and Grouping Solutions by Using Rotated Copies", (submitted to Journal of Pattern Recognition).

[5] **L. Idoumghar, D. Fodorean and A. Miraoui**, "Electrical machine optimization based on hybrid evolutionary algorithm", (submitted to Journal of Energy Conversion & Management, Elsevier).

32 rue de Ruelisheim, Illzach – 68110

+33 6 99 90 70 33 • +33 3 89 33 60 25 • +33 3 89 33 64 91

• lhassane.idoumghar@uha.fr • www.lmia.uha.fr/~mage/idoumghar/

[6] **L. Idoumghar, M. Idrissi-Aouad, M. Melkemi and R. Schott**, "Particle Swarm Optimization Algorithm with Mutation Operator For Multi-Modal Function Optimization and Reducing Energy Consumption in Embedded Systems", (submitted to Journal of Applied Mathematics and Computation).

Articles de vulgarisation scientifique

Proposition de plusieurs sujets pour les épreuves TIPE pour les Concours Communs Polytechniques. Des sujets tels que :

[1] **L. Idoumghar**, "Diagramme de Voronoï 3D", pp. 1-15, 2006.

[2] **L. Idoumghar**, "Gestion des Flux", pp. 1-12, 2007.

[3] **L. Idoumghar**, "N'aimez-vous pas prévoir le futur parfois (chaine de Markov) ?".

[4] **L. Idoumghar**, "Métaheuristique pour les problèmes d'optimisation combinatoire", pp. 1-13, 2010.

[5] **L. Idoumghar**, "Utilisation des essaims particulaires (PSO) pour l'apprentissage de Modèles de Markov Cachés", pp. 1-14, 2011.

Logiciels

[1] J'ai développé une application pour étudier le comportement des métahéuristiques sur des benchmarks mathématiques. Cette application est dotée de deux interfaces graphiques (2D et 3D). Elle est téléchargeable sur le lien suivant :

<http://www.lmia.uha.fr/~mage/idoumghar/MetaheuristicDemo.jar>

[2] J'ai développé, pour TDF, un logiciel d'allocation de fréquences, ALOES (ALgorithme d'Optimisation de l'Emploi de Spectre), actuellement utilisé pour le déploiement en France et en Europe de la Télévision Numérique par voie Terrestre (TNT).

Évaluation d'articles pour des conférences ou des revues

Rapporteur pour les conférences IASTED (AIA, ASC), Journal of Swarm and Evolutionary Computation, Journal of IEEE Transactions on Vehicular Technology, Journal of IEEE Transactions on Magnetics, Journal of Engineering Applications of Artificial Intelligence, Journal of Electrical Engineering in Transportation, Journal of Algorithms.

Mes principales collaborations pour le prochain contrat quinquinal

- **D. Fodorean**, Cluj-Napoca University-Romania : métahéuristiques pour la modélisation et optimisation des machines électriques.
- **A. Abbou**, Ecole Mohammadia d'Ingénieurs-Rabat Maroc : métahéuristiques pour l'optimisation d'un moteur électrique.
- **A. Miraoui**, Université de Technologie Belfort-Montbéliard : algorithmes hybrides pour l'optimisation de la gestion de l'énergie dans les smarts-grid.

- **R. Schott**, INRIA-Grand Est-Nancy : optimisation multi-objectifs pour la planification des réseaux de radiodiffusion.
- **Ph. Debreux et Th. Schott**, TDF-Metz : optimisation multi-objectifs pour les réseaux de radio-diffusion.
- **A. Abouaissa**, MIPS-UHA Mulhouse : métaheuristiques pour le déploiement d'un réseau de capteurs.
- **M. Rekik**, Université Laval-Québec : mise en œuvre d'algorithmes d'affectation de ressources humaines à des projets en tenant compte des conventions collectives.
- **P. Collet** LSiiT-UDS Strasbourg et **J.-L. Paillaud**, IS2M-UHA Mulhouse : développement d'une métaheuristique massivement parallèle pour la résolution structurale d'une nouvelle zéolite.
- **M. Melkemi**, LMIA-UHA Mulhouse : mise en place des algorithmes géométriques pour la détection de formes vides dans des nuages de points.
- **M. Idrissi**, Ecole IGA-Casablanca, Maroc : développement de métaheuristiques hybrides pour la gestion de l'emploi du temps pour les universités.
- **M. Benkherrat**, Ecole d'ingénieurs EPMI-Cergy Pontoise : métaheuristiques hybrides pour l'extraction du bruit des enregistrements neurophysiologiques individuels par modélisation adaptative basée sur le polynôme de l'Hermite.
- **Ch. Guyeux**, AND-LIFC Besançon : métaheuristiques pour le repliement de protéines.

Responsabilités et autres activités

Depuis 2009 **Responsable pédagogique de la Licence Professionnelle "Réseaux et Télécommunications" en formation continue**, *Université de Haute Alsace*.

En 2009, j'ai été sollicité pour prendre la responsabilité de la Licence Professionnelle "Réseaux et Télécommunications" en formation continue à l'UHA. Depuis, j'ai accepté de reprendre le flambeau de cette licence sur le plan administratif et pédagogique, et je prends en charge une partie des enseignements de cette formation. Cette responsabilité implique une participation active dans le recrutement des étudiants (jury de sélection), la préparation des jurys pour les soutenances de stages et des projets ainsi que la mise en place des jurys de fin d'année. Sans oublier le suivi des stages des étudiants durant le second semestre de l'année universitaire. Cette responsabilité implique aussi l'animation de l'équipe pédagogique, le recrutement des chargés de cours-TD pour les enseignements en vacations, ainsi que la coordination des différentes matières. Une demande de renouvellement de l'habilitation est déposée pour le prochain contrat quinquennal.

Depuis 2007 **Responsable pédagogique pour la partie informatique premier semestre**, *Université de Haute Alsace*.

Depuis 2007, je suis responsable pédagogique pour la partie informatique pour le premier semestre. Cette responsabilité implique la mise en place et la vérification de l'emploi du temps pour les différentes matières informatiques du premier semestre. Sans oublier l'animation de l'équipe informatique et la coordination de l'avancement des différents cours et les travaux pratiques.

- Depuis 2003 **Examinateur, TIPE - Concours Communs Polytechniques (CCP).**
 Depuis 2003, je suis examinateur pour les TIPE des Concours Communs Polytechniques (CCP) et depuis 2010 je suis responsable pédagogique adjoint, pour la branche Mathématiques et Physique. Cette responsabilité implique de proposer (et de "rapporter" sur des sujets) pour le concours TIPE. Sans oublier la participation à la coordination des différents jurys et la vérification du bon déroulement des épreuves.
- 11 – 13 Avril 2012 **Organisation de deux sessions invitées “Métaheuristiques hybrides”, 13^{ème} congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2012), Angers.**
- 2011 – 2012 **J'ai participé au montage de la fédération SPI-STIC-Math,** Je suis co-porteur du projet MIFA "Mathématique de l'Image, des Formes et Applications". La fédération SPI-STIC-Math regroupera 3 laboratoires de l'UHA : LMIA, MIPS et LPMT. L'objectif du projet MIFA est de mettre en commun les compétences des chercheurs appartenant aux trois laboratoires de la fédération, et améliorer ainsi la synergie entre les différentes équipes de recherche.
- 2 – 4 Mars 2011 **Organisation de deux sessions invitées “Métaheuristiques hybrides”, 12^{ème} congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2011), Saint-Etienne.**
- 27 – 29 Octobre 2010 **Présidence de la session “Swarm intelligence”, 22th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2010), Arras.**
- 26 – 30 janvier 2009 **Participation à l'organisation des Journées de Géométrie Algorithmique,** Mulhouse.
- 12 – 14 Mars 2008 **Participation à l'organisation de l'École de printemps du GDR Informatique Graphique,** Vosges.
- 2006-2012 **Participation au montage d'un Master M2 recherche,** Cohabilitation UHA, UTBM et UFC.
 Ne pouvant pas avoir d'étudiants en M2 spécialité recherche (suite aux faibles effectifs en M1 informatique à l'UHA jusqu'en 2007) nous avons pris la décision de monter, en 2006, le dossier pour le M2 spécialisé en "Algorithmique, Hautes Performances et Modélisation" (spécialité recherche). Ce M2 est cohabilité entre l'Université de Haute-Alsace, l'Université de Franche-Comté et l'Université de Technologie de Belfort-Montbéliard.
- Depuis 2010 **Participation au montage d'une nouvelle spécialité M2 à la fois professionnelle et recherche dans le domaine de “l’Informatique Mobile et Répartie”.**
 En 2010, j'ai participé au montage d'une nouvelle spécialité M2 à la fois professionnelle et recherche dans le domaine de "l’Informatique Mobile et Répartie". Cette spécialité sera cohabilité entre l'Université de Haute Alsace, l'Université de Franche-Comté et l'Université de Technologie de Belfort-Montbéliard. Ce master sera ouvert dès la rentrée scolaire 2012/2013.
- 2008 – 2012 **Membre élu au Conseil Scientifique,** Université de Haute Alsace.
- 2012 **Membre du Comité de Sélection,** Université de Haute Alsace, Poste d'Informatique à la FST.
- 2009 – 2011 **Membre de la Commission de réflexion sur les modalités d'attribution de la Prime d'Excellence Scientifique,** Université de Haute Alsace.
- 2011 **Membre de la Commission de réflexion sur la procédure HDR,** Université de Haute Alsace.

- 13 Mars 2008 **Présentation lors des Journée de l'École de printemps du GDR Informatique Graphique**, Mulhouse.
- Depuis 2004 **Organisation des séminaires au sein du LMIA**, Mulhouse.
- 19 Février 2004 **Séminaire à l'Université de Haute Alsace**, Mulhouse.
- 08 Avril 2004 **Participation et présentation lors de la journée CRESPIM**, Mulhouse.
- 25 Avril 2003 **Séminaire à l'Université Paris XI, Laboratoire LRI, Equipe théorie des graphes**, Paris.
- 2002 **Participation et présentation lors de la journée PDMIA (Processus Décisionnels de Markov et Intelligence Artificielle)**, Nancy.
- 2001 – 2002 **Organisation des séminaires au sein de TDF-C2R**, Metz.
- 2010 **Participation à la journée JFPDA'10**, Besançon.
- 2002 **Séminaire à l'Institut National de Polytechnique de Lorraine (INPL), Equipe TRIO**, au LORIA, Nancy.
- Depuis 2011 **Membre ROADEF**, (*Recherche Opérationnelle et d'Aide à la Décision*).
- Depuis 2008 **Membre GTMG**, (*Groupe de Travail en Modélisation Géométrique*).
- Depuis 2001 **Membre du groupe PDMIA**, (*Processus Décisionnels de Markov et Intelligence Artificielle*).

Enseignements

- 2007 – 2012 **Optimisation et calcul distribué**, *Cours, TD et TP*, Master 1 Informatique, [20 HTD].
J'ai la responsabilité du cours d'optimisation en M1 informatique et Miage. Les concepts abordés sont les métahéuristiques et le calcul distribué. Ce cours est l'occasion d'étudier certaines métahéuristiques et permet une ouverture vers leurs utilisations, au niveau recherche, pour résoudre certains problèmes d'optimisation (exemple : Affectation quadratique, coloriage de graphe, voyageur de commerce, emploi du temps). Ce cours aborde également une introduction au calcul distribué et l'étude des performances des algorithmes séquentiels et distribués mis en place lors des séances de TP. Les versions distribuées sont implémentées en utilisant MPI (*Message Passing Interface*).
- 2007-2012 **Algorithmique**, *Cours et TD*, Licence 1 année, [84 HTD].
Le but de ces C-TD est de présenter aux étudiants de L1 une méthodologie de résolution de problèmes en informatique. Dans cette optique, une introduction à l'algorithmique est présentée ainsi que différents concepts essentiels en informatique : structures de contrôle, boucles, fonctions, procédures et tableaux

2007 – 2012 **Langage C++, Cours, TD et TP**, Licence 1 année, [48 HTD].

Le cours sur langage C++ est destiné aux étudiants de L1 Mathématiques et Informatique. Le but de ce cours est d'apprendre la syntaxe de C++ et d'approfondir l'aspect analyse souvent marginalisé par les étudiants. En collaboration avec B. Adam, nous proposons des projets aux différents groupes pour mettre en pratique les différentes notions vues en C-TD et en TP. Ces projets sont particulièrement motivants, de par l'implication qu'ils demandent mais également en raison de l'implication des étudiants eux-mêmes. Noter que la partie graphique des projets proposés est réalisée à l'aide de la bibliothèque "winbgi".

2008 – 2012 **Outils Informatique, TD**, Licence 1 année, [49 HTD].

Cette matière vise à donner aux étudiants de L1 la maîtrise des outils de traitement de texte, de tableur, ainsi qu'à les initier à l'utilisation d'internet. L'accent est mis sur les possibilités de liaison entre les différents logiciels (traitement de texte, tableur, navigateur). Les étudiants apprennent également à affiner leurs recherches de l'information sur le web, à mettre en forme des documents ainsi qu'à utiliser les fonctions de calcul d'un tableur (calcul de somme, de moyennes, liaison entre les feuilles de calcul, graphiques).

2009 – 2012 **Cours de Java, Cours, TD et TP**, L3 Informatique, [35 HTD].

Le but de ce cours est d'apprendre la syntaxe de Java et d'approfondir l'aspect analyse souvent marginalisé par les étudiants. Dans un deuxième temps, j'aborde une introduction à la programmation orientée objet à l'aide du langage Java.

2008 – 2012 **Algorithmes évolutionnaires et projet transversal, Cours et TP**, Master 2 recherche-Informatique, [18 HTD].

Dans le M2 AHPM (Algorithmique, Hautes Performances et Modélisation) qui est en cohabitation avec l'UTBM et l'UFC, je suis chargé d'assurer un cours sur les algorithmes évolutionnaires et de proposer aux étudiants un projet transversal qui fait appel aux thèmes des équipes qui interviennent dans ce master recherche. Le déroulement de cette matière est le suivant : chaque année, une ou deux métaheuristiques sont étudiées en cours avec illustration sur des exemples. Il est ensuite demandé aux étudiants de mettre en pratique l'approche étudiée pour résoudre un problème extrait des récents articles de recherche. Les trois derniers projets proposés sont :

- Projet 2011 - 2012 : le but de ce projet est de mettre en place une métaheuristique hybride (Essaim particulaire combiné avec l'algorithme génétique) pour l'optimisation continue. Chaque étudiant doit fournir une version séquentielle et une version distribuée de son programme. Cette dernière version doit être réalisée à l'aide de MPI.
- Projet 2010 - 2011 : le but de ce projet est de mettre en place une métaheuristique hybride (Essaim particulaire combiné avec le recuit simulé) pour résoudre le problème de modélisation d'un transistor. Chaque étudiant doit fournir une version séquentielle et une distribuée de son programme. Cette dernière version doit être réalisée à l'aide de Java RMI.
- Projet 2009 - 2010 : ce projet se décompose en deux parties.
 - Mise en place d'un algorithme séquentiel basé sur l'optimisation par essaim particulaires (Particle Swarm Optimisation) pour résoudre le problème d'affectation quadratique. La programmation doit se faire en utilisant Java ou C++.
 - Parallélisation d'un algorithme génétique que j'ai développé pour résoudre le problème N-Reines. La programmation doit se faire en utilisant Java RMI.
- Projet 2008 - 2009 : Le but de ce projet est de résoudre le problème d'affectation quadratique en utilisant la méthode tabou. Chaque étudiant devrait proposer une version séquentielle et distribuée de la méthode tabou. La version distribuée devrait fonctionner sous les deux plateformes Linux et Windows. Elle devrait aussi utiliser la bibliothèque MPI.

2007 – 2011 **Système d'exploitation, Cours et TP**, L3 en formation continue, [13 HTD].

Dans ce cours, il s'agit, après une introduction au principe de fonctionnement d'un système d'exploitation, d'étudier la notion de processus (définition, Mise en œuvre, Communication, Ordonnancement), la gestion de la mémoire (Mémoire sans va-et-vient ni pagination, Le va-et-vient, La mémoire virtuelle). Dans ce cours une introduction au shell Unix (Bourne Shell) et aux scripts (variables, tests, boucles, fonctions) a été abordée.

32 rue de Ruelisheim, Illzach – 68110

+33 6 99 90 70 33 • +33 3 89 33 60 25 • +33 3 89 33 64 91
• lhassane.idoumghar@uha.fr • www.lmia.uha.fr/~mage/idoumghar/

2007 – 2011 **Architecture des ordinateurs**, *Cours et TD*, L3 en formation continue, [11 HTD].

Ce cours a pour but d'introduire différentes notions nécessaires à la compréhension de l'Architecture d'un ordinateur. Dans ce cours, j'ai traité la représentation des données (Représentation des caractères, des nombres entiers / réels (norme IEEE)) et l'architecture de base d'un ordinateur (fonctionnement, la mémoire (vive, morte, cache), les bus, architecture des microprocesseurs), Architecture d'un Dual-Core.

Résumé & Abstract

Résumé

La résolution des problèmes d'optimisation difficiles requiert la mise en place de métaheuristiques, qui sont une famille d'algorithmes stochastiques. Ces algorithmes, combinés souvent avec des méthodes de recherche locale, permettent d'obtenir des solutions de très bonne qualité dans un temps raisonnable.

Dans ce mémoire d'Habilitation à Diriger des Recherches, je synthétise une partie de mes contributions à la résolution informatique de certains problèmes industriels et académiques que l'on peut qualifier de problèmes difficiles. Je montre, notamment, comment l'hybridation entre des métaheuristiques (ou avec des systèmes multi-agents) nous a permis d'obtenir des solutions de très bonne qualité dans des domaines tels que l'allocation de fréquences en radiodiffusion, la gestion intelligente de l'énergie dans les *smart grids*, la modélisation d'un moteur électrique à aimant permanent, la réduction de l'énergie consommée en mémoire dans les systèmes embarqués et en optimisation continue.

A la fin du mémoire, je présente mes perspectives de recherche, en détaillant quelques pistes pour la définition de nouvelles approches génériques pour la résolution de problèmes réels de grande taille.

Mots clés : métahéuristiques hybrides, système multi-agents, optimisation continue, allocation de fréquences, *smart grids*, moteur électrique à aimant permanent, mémoire *Scratch-Pad*.

Abstract

Solving hard optimization problems requires the implementation of metaheuristics, which are a family of stochastic algorithms. These algorithms, often combined with local search methods, have the advantage of obtaining high quality solutions in a reasonable time.

In this accreditation to supervise research, I summarize my contributions for solving some industrial and academic hard problems. For example, I show how hybridization between metaheuristics (or with multi-agent systems) has allowed us to obtain high quality solutions in many areas such as frequency assignment in the field of radio-broadcasting, intelligent energy management in *smart grids*, modeling of a permanent magnet electric motor, reducing the energy consumed in memory in embedded systems, etc.

At the end of this dissertation, I present my research perspectives as well as some ideas for the definition of generic approaches for solving real and large size problems.

Keywords : hybrid metaheuristic, multi-agent systems, continuous optimization, frequency assignment problem, *smart grids*, permanent magnet electric motor, Scratch-Pad memory.

Resumé

Introduction générale

1 Historique de mes activités

Il me paraît très utile de présenter un petit historique de mon parcours en insistant sur les différentes étapes qui ont marqué l'évolution de mes activités de recherche, avant de détailler le contexte et les objectifs de nos travaux.

1.1 Thèse de doctorat CIFRE (01/02/2000-12/12/2002)

Ma thèse CIFRE a été réalisée dans le Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA) en partenariat avec TéléDiffusion de France (TDF). Son objectif était de mettre en place des méthodes algorithmiques pour la génération des plans de fréquences dans le cadre de la radiodiffusion (réseau FM et Télévision Numérique Terrestre par voie Hertzienne). Cette génération est une tâche difficile dans le processus de planification, qui peut conduire à des solutions peu adéquates d'un point de vue métier. En effet, le processus de génération s'appuie, d'une part, sur une modélisation des contraintes existant entre les points de service du réseau étudié, et d'autre part sur une optimisation combinatoire, qui vise à satisfaire ces contraintes. Cette optimisation combinatoire fournit une solution optimale d'un point de vue mathématique, mais selon la finesse de modélisation des contraintes, la solution générée peut être inutilisable dans la réalité. Dans cette thèse, nous avons développé de nouvelles méthodes algorithmiques permettant de résoudre efficacement le problème d'allocation de fréquences dans le cadre de la radiodiffusion. L'utilisation de ces nouvelles approches a montré que la qualité des solutions obtenues est nettement supérieure à celle des meilleures solutions opérationnelles existantes dans ce domaine. TDF, fort de ces résultats, utilise actuellement ces approches dans

le cadre du déploiement de la TNT en France, ainsi qu'en Europe (via ses filiales).

1.2 Poste ATER et Maître de conférences à l'Université de Haute Alsace, Mulhouse

Après ma thèse, j'ai effectué une année (2003-2004) d'ATER à l'université de Haute Alsace (UHA) avant d'y être recruté sur un poste de Maître de Conférences en septembre 2004. Mon laboratoire de rattachement, "Modélisation et Algorithmique GÉométrique" (MAGE), a fusionné en 2004 avec le laboratoire de Mathématiques pour former le Laboratoire de Mathématiques, Informatique et Applications (LMIA) composé de deux équipes de Mathématiques et une équipe d'informatique, qui a gardé l'appellation MAGE. L'axe de recherche principal de MAGE, mon équipe d'accueil, est la géométrie algorithmique. Un second axe, "algorithmique générale", a été développé par le professeur Spehner avant son départ à la retraite en 2004. Sachant que mes thématiques de recherche s'inscrivent uniquement dans l'axe secondaire, j'avais la responsabilité de développer seul une nouvelle thématique, tout en travaillant avec mes collègues de la géométrie algorithmique. Durant le dernier contrat quadriennal 2008-2012 :

- j'ai renforcé l'axe de recherche "algorithmique générale" (baptisé "métaheuristiques et optimisation combinatoire" pour le prochain contrat 2013-2018) en développant des collaborations académiques (INRIA-TRIO Nancy, UTBM Belfort, Université de Cluj Roumanie, Ecole Mohammadia d'Ingénieurs, Maroc) et industrielles (General Electric, TéléDiffusion de France, ESDI). Mes principaux travaux de recherche dans ce domaine se sont orientés vers l'utilisation des métaheuristiques hybrides pour la résolution de problèmes industriels. Suite aux résultats que j'ai obtenus et à la dynamique des collaborations que j'ai mises en place, un Maître de Conférences est recruté cette année pour renforcer l'axe "métaheuristiques et optimisation combinatoire" qui est sous ma responsabilité. Les résultats de l'évaluation très positive de cet axe par l'AERES me confortent à continuer mes futurs travaux de recherche dans ce domaine durant le prochain contrat quinquennal (2013-2018).
- sur l'axe géométrie algorithmique, j'ai développé, en collaboration avec le professeur Melkemi, des travaux ayant pour but la mise en place de méthodes d'optimisation (à base de concepts géométriques) pour la détection de formes vides dans un nuage de points. Ces travaux ne seront pas présentés dans ce mémoire, pour garder une certaine cohérence des thématiques abordées (le lecteur intéressé trouvera deux articles complets sur ce sujet à la fin de ce mémoire).

2 Contexte et objectifs

De nombreux problèmes, d'ordre pratique ou théorique, peuvent être modélisés sous forme de problème d'optimisation. Que ce soit pour l'allocation de fréquences en radiodiffusion ou radiocommunication, la reconnaissance d'objets dans des images, que ce soit pour réguler et contrôler la gestion de la production électrique, ou bien pour préciser les paramètres et les dimensions d'un moteur électrique, nous pouvons être amenés à résoudre des problèmes d'optimisation, puisque nous cherchons le plus souvent à maximiser une performance ou à minimiser un coût. D'une manière générale, un problème d'optimisation est défini par un ensemble d'instances. À chaque instance du problème est associé un ensemble de solutions S , un sous-ensemble X de S représentant les solutions admissibles/réalisables et une fonction, appelée fonction de coût ou objectif, qui affecte à chaque solution admissible une valeur scalaire. Lorsque l'ensemble des solutions admissibles est fini nous parlons d'*optimisation combinatoire*. Résoudre une instance du problème consiste à trouver une solution admissible dont la valeur scalaire associée est maximale ou minimale, suivant qu'il s'agit, respectivement, d'un problème de maximisation ou de minimisation. Une telle solution est appelée *solution optimale*. Un algorithme d'optimisation pour résoudre le problème est un algorithme qui, pour chaque instance, permet de trouver une solution optimale. Lorsque nous cherchons à optimiser simultanément plusieurs objectifs, nous nous intéressons à des problèmes d'optimisation multiobjectif.

Les problèmes traités dans ce mémoire sont principalement des problèmes d'optimisation industriels réputés difficiles. Il en résulte que nous ne connaissons pas de méthodes de résolution exactes permettant de trouver la solution optimale en un temps raisonnable, pour des instances du problème de grandes tailles. Il devient alors nécessaire de proposer des algorithmes approchés fournissant, en temps acceptable, des solutions satisfaisantes. Un tel procédé ad hoc appliqué à un problème spécifique est appelé une *heuristique*. Lorsque la méthode de résolution est exprimée à un niveau abstrait, indépendant du problème d'optimisation considéré, elle est appelée *métaheuristique*. L'évaluation de ces méthodes de résolution (métaheuristiques et heuristiques) se fait généralement de manière expérimentale et elles sont validées par l'évidence statistique. Les outils formels existants ne permettent généralement pas de garantir théoriquement le résultat obtenu en pratique.

Pour résoudre les problèmes d'optimisation que nous avons étudiés, nous nous sommes intéressés à l'hybridation de concepts et de méthodes qui proviennent de différents domaines de l'intelligence artificielle et de la recherche opérationnelle. Ces nouvelles méthodes, dites hybrides, consistent à combiner des méthodes exactes et/ou des algorithmes approchés pour tenter de tirer profit des points forts de chaque composante et améliorer le comportement

global de l'algorithme ainsi conçu. Par exemple, nous serons amenés à combiner des approches évolutionnaires avec des recherches locales, et ensuite à analyser leur conception sous l'angle de la résolution collective de problèmes. Se pose alors la question : comment hybrider/combiner des approches, a priori différentes, pour former un algorithme de résolution de problèmes efficace et qui englobe de manière cohérente ces concepts variés ? La réponse à cette question constitue une préoccupation sous-jacente aux travaux présentés dans ce mémoire. De manière générale, nous cherchons à proposer des méthodes hybrides :

- *robustes et performantes*. En temps raisonnable, ces méthodes doivent fournir des solutions optimales ou quasi-optimales.
- *dynamiques et stochastiques*. Ces méthodes doivent offrir la possibilité d'adaptation aux problèmes dans lesquels la fonction objectif ou les contraintes sont fonctions du temps.
- *séquentielles et distribuées*. L'implémentation séquentielle est souvent à exécuter dans un contexte centralisé, contexte dans lequel le système qui résout est différent du système physique modélisé. La parallélisation de ces méthodes hybrides permet souvent des gains de temps.
- *modulables et généralisables* à d'autres types d'applications. Comme pour les métahéuristiques traditionnelles, ces nouvelles méthodes hybrides auront l'avantage de ne pas dépendre du problème traité. Ceci est d'autant plus vrai qu'elles utilisent des métaphores naturelles/biologiques et manipulent des entités abstraites.

L'objectif de ces méthodes hybrides, à plus ou moins long terme, est d'étudier comment l'utilisation de métaphores biologiques peut apporter la robustesse au regard d'approches traditionnelles, telle que, par exemple, les méthodes heuristiques itératives de la recherche opérationnelle. D'autre part, il s'agit d'étudier l'implémentation parallèle de ces méthodes, sur des stations de travail en réseau ou sur des systèmes multi-processeurs, ou sur les différents dispositifs physiques définissant le problème (en tant que supports d'exécution eux-mêmes) dans la perspective de résolution de problèmes de grandes tailles.

3 Plan du mémoire

Nous présentons nos travaux en six chapitres principaux dont nous avons voulu qu'ils reflètent cinq types de problèmes (industriels et académiques) que nous avons traités. Pour les cinq derniers chapitres, nous présentons à chaque fois une description du problème, ensuite nous décrivons les approches conçues pour résoudre le problème, ainsi que les résultats expérimentaux obtenus.

Chapitre I - État de l'art sur les métaheuristiques

Ce premier chapitre est consacré à l'étude bibliographique des métaheuristiques tout en décrivant certaines méthodes à solution unique ou à base de population. De même, nous introduisons les méthodes hybrides qui combinent les algorithmes évolutionnaires avec la recherche locale, ainsi que les modèles parallèles associés.

Chapitre II - Métaheuristiques pour la planification des réseaux de radiodiffusion

Après une introduction du problème d'allocation de fréquences en radiodiffusion, nous présentons dans ce chapitre les deux modélisations que nous avons proposées pour résoudre ce problème. Ensuite, après la description des métaheuristiques développées, nous donnerons les résultats de la validation métier des résultats obtenus par nos approches.

Chapitre III - Algorithmes hybrides pour la gestion intelligente de l'énergie dans les *smart grids*

Nous nous intéressons dans ce chapitre à l'hybridation des systèmes multi-agents et des métaheuristiques dans le domaine de l'optimisation dynamique. Pour cela, quelques définitions sur les systèmes multi-agents sont énoncées, puis nous présentons les résultats obtenus pour la gestion de la production électrique en combinant différentes sources d'énergie (turbines à gaz, éoliennes, photovoltaïques, réseau électrique classique, etc.). En plus d'apporter une validation expérimentale de nos approches, nous cherchons à montrer leur facilité d'adaptation à différents problèmes d'optimisation dynamique.

Chapitre IV - Métaheuristiques pour la réduction de l'énergie consommée en mémoire dans les systèmes embarqués

Nous considérons dans ce chapitre la problématique liée à la gestion mémoire basse énergie dans les systèmes embarqués. Après une présentation de nos métaheuristiques hybrides, nous confrontons nos résultats expérimentaux à des approches de résolution récemment publiées dans ce domaine.

Chapitre V - Métaheuristiques pour l'optimisation de moteurs électriques à aimant permanent

Ce chapitre propose une autre application de nos métaheuristiques sur un problème de dimensionnement d'un moteur électrique. Après une introduction au problème de la conception d'un moteur électrique à aimant permanent, ce chapitre mettra en avant l'intérêt

Introduction générale

d'utiliser des métaheuristiques dans la conception d'un moteur électrique puisque nos approches ont permis de trouver les bons dimensionnements du moteur électrique demandé par le cahier des charges.

Chapitre VI - Métaheuristiques pour l'optimisation continue

Dans ce chapitre, nous proposons une validation de nos algorithmes d'optimisation sur une série de *benchmarks* mathématiques. La méthode de validation est basée sur l'utilisation du test de Friedman ([Garc 09]). Les résultats montrent que nos algorithmes obtiennent de bons résultats en comparaison de plusieurs approches récemment publiées.

Chapitre VII - Conclusion générale et perspectives

Le mémoire se termine par un bilan des travaux de recherche. Nous dressons un ensemble de perspectives pouvant être développées par la suite, ouvrant ainsi de nouvelles voies et propositions dans le but d'améliorer ce travail.

Chapitre I

État de l'art sur les métaheuristiques

1 Vocabulaire et définitions

Dans cette section, nous donnons un ensemble de définitions de certains termes utilisés couramment dans le domaine de l'optimisation mono-objectif et multiobjectif.

Définition 1

Le **vecteur de décision** \vec{x} représente l'ensemble des variables du problème.

Définition 2

L'**espace de recherche** représente l'ensemble des valeurs pouvant être prises par les variables de décision, on le représente par χ .

Définition 3

L'**espace réalisable** représente l'ensemble des valeurs des variables satisfaisant les contraintes du problème.

Définition 4

Un point $\vec{x} \in \chi$ est un **minimum global** du problème si et seulement si : $\forall \vec{x} \in \chi, f(\vec{x}) \leq f(\vec{x}')$.

Définition 5

Un point $\vec{x} \in \chi$ est un **minimum local** du problème si et seulement si : $\forall \vec{x}' \in \mathcal{V}(\chi), f(\vec{x}) \leq f(\vec{x}')$, où $\mathcal{V}(\chi)$ définit un “voisinage” de \vec{x} .

Définition 6

Soient **u** et **v**, deux vecteurs de même dimension,

$$\begin{aligned}\mathbf{u} = \mathbf{v} &\quad \text{ssi } \forall i \in \{1, 2, \dots, m\}, u_i = v_i \\ \mathbf{u} \leq \mathbf{v} &\quad \text{ssi } \forall i \in \{1, 2, \dots, m\}, u_i \leq v_i \\ \mathbf{u} < \mathbf{v} &\quad \text{ssi } \mathbf{u} \leq \mathbf{v} \wedge \mathbf{u} \neq \mathbf{v}\end{aligned}$$

Les relations \geq et $>$ sont définies de manière analogue.

Chapitre I. État de l'art sur les métaheuristiques

Définition 7

La dominance au sens de Pareto : Considérons un problème de minimisation. Soient \mathbf{u} et \mathbf{v} deux vecteurs de décision :

$$\begin{array}{ll} \mathbf{u} \prec \mathbf{v} \text{ (\mathbf{u} domine \mathbf{v})} & \text{ssi } \mathbf{f}(\mathbf{u}) < \mathbf{f}(\mathbf{v}) \\ \mathbf{u} \preceq \mathbf{v} \text{ (\mathbf{u} domine faiblement \mathbf{v})} & \text{ssi } \mathbf{f}(\mathbf{u}) \leq \mathbf{f}(\mathbf{v}) \\ \mathbf{u} \sim \mathbf{v} \text{ (\mathbf{u} est incomparable (ou non-dominé) avec \mathbf{v})} & \text{ssi } \mathbf{f}(\mathbf{u}) \not\leq \mathbf{f}(\mathbf{v}) \wedge \mathbf{f}(\mathbf{v}) \not\leq \mathbf{f}(\mathbf{u}) \end{array}$$

Définition 8

Une solution $x \in \chi_f$ est dite non dominée par rapport à un ensemble $\chi_a \subseteq \chi_f$ si et seulement si :

$$\nexists x_a \in \chi_a, x_a \prec x$$

Définition 9

Un vecteur de décision $x \in \chi$ est dit **Pareto globalement optimal** si et seulement si : $\nexists y \in \chi, y \preceq x$. Dans ce cas, $f(x) \in \mathcal{F}$ est appelée solution efficace.

Définition 10

Un vecteur de décision $\mathbf{x} \in \chi$ est dit **Pareto optimal localement** si et seulement si, pour $\delta > 0$ fixé : $\nexists y \in \chi, f(y) \in B(f(x), \delta)$ et $y \prec x$, où $B(f(x), \delta)$ représente une boule de rayon $f(x)$ et de rayon δ .

2 Problème d'optimisation mono-objectif

2.1 Définition

Nombreux sont les problèmes d'optimisation rencontrés dans l'industrie qui ont pour but de chercher une ou plusieurs solutions qui optimisent une fonction objectif. Ces problèmes d'optimisation mono-objectif peuvent être formulés de la manière suivante :

$$\left\{ \begin{array}{l} \text{optimiser } f(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R} \\ \text{Tel que : } \vec{C}(\vec{x}) \leq 0 \\ \text{Avec : } \vec{x} \in \mathbb{R}^n, \vec{C} \in \mathbb{R}^q \end{array} \right. \quad (I.1)$$

où f est la fonction objectif, \vec{x} est un vecteur de variables de décision, C un ensemble de contraintes que doit satisfaire une solution afin d'être admissible.

I.2 Problème d'optimisation mono-objectif

La résolution du problème (I.1) nécessite de parcourir une partie de l'*espace de recherche*¹ pour trouver une solution appartenant à l'*espace de solutions réalisables*².

Dans la littérature, deux types d'approches sont souvent mises en opposition pour résoudre les problèmes d'optimisation : les méthodes déterministes (ou “exactes”) et les méthodes approchées.

2.2 Méthodes exactes

On trouve dans la bibliographie de nombreuses méthodes exactes ([Dhae 05], [Bari 03], [Hao 99], [Help 70], [Beas 93], [Glov 65], [GLOV 77]) qui permettent de résoudre certains types de problèmes. Ces méthodes nécessitent que la fonction objectif présente un certain nombre de caractéristiques, telles que la convexité, la continuité ou encore la dérivabilité. Parmi les méthodes les plus connues, on peut citer par exemple :

- La méthode de *séparation et évaluation* (*branch-and-bound*). Il s'agit d'une méthode d'énumération implicite : toutes les solutions possibles du problème peuvent être énumérées, mais l'analyse des propriétés du problème permet d'éviter l'énumération de larges classes de mauvaises solutions. De manière générale, dans un bon algorithme par séparation et évaluation, seules les solutions potentiellement bonnes sont donc énumérées. L'algorithme *branch and bound*, dédié aux problèmes de recherche opérationnelle, est souvent comparé à l'algorithme *A** qui, lui, est très souvent utilisé en intelligence artificielle.
- La méthode avec *retour arrière* (*backtracking*). Il s'agit d'une méthode de construction de solution de manière itérative. Elle se base sur un parcours en profondeur ou en largeur de l'espace des solutions représenté sous forme d'un arbre. Cette méthode permet, à partir d'une solution partielle, de dire assez tôt si une décision va aboutir ou pas. Dès qu'un échec est détecté, la méthode revient sur une décision déjà prise (*backtrack*) pour explorer une nouvelle décision (nouvelle solution).
- La *programmation dynamique*. Elle permet de résoudre des problèmes d'optimisation dont la fonction objectif se décrit comme la somme de fonctions monotones non-décroissantes des ressources. Plus précisément, son principe consiste à placer le problème étudié dans une famille de problèmes (sous-problèmes) moins difficiles, mais de même nature, et ainsi on doit pouvoir déduire la solution optimale du problème de départ à partir d'une solution optimale d'un sous-problème.

1. Ensemble des valeurs pouvant être prises par les variables du problème traité

2. Ensemble des valeurs des variables satisfaisant les contraintes liées au problème traité

Chapitre I. État de l'art sur les métaheuristiques

Dans la littérature, plusieurs bibliothèques génériques existent et permettent la résolution des problèmes pouvant s'écrire sous une forme algébrique en variables entières, ou binaires, ou mixtes. Parmi ces bibliothèques, on peut citer : CPLEX, MPL, AMPL, OMP, LINDO, *etc.*

L'avantage des méthodes exactes est qu'elles permettent d'obtenir une ou plusieurs solutions dont l'optimalité est garantie mais elles ont généralement le défaut d'être très coûteuses en temps de calcul pour des problèmes difficiles. En effet, le temps nécessaire pour trouver une solution risque d'augmenter exponentiellement avec la taille du problème. Sans oublier que certaines caractéristiques (la discontinuité, la non-dérivabilité ou encore la présence de bruit) peuvent être problématiques pour ces méthodes. Dans ce cas, le problème d'optimisation est dit *difficile*, car aucune méthode déterministe ne peut résoudre ce problème en un temps raisonnable. Dans les problèmes d'optimisation difficile, on distingue deux types de problèmes :

- Les problèmes d'optimisation à variables discrètes (ou combinatoire) qui consistent à trouver, dans un ensemble discret, la meilleure solution réalisable. Par exemple, on trouve le célèbre problème du voyageur de commerce, qui doit visiter un certain nombre de villes, avant de retourner à la ville de départ. Pour ce type de problème, l'inconvénient majeur réside dans le fait que le nombre de solutions réalisables est généralement très élevé, donc il est très difficile de trouver la meilleure solution dans un temps raisonnable.
- Les problèmes à variables continues, qui sont moins formalisés que les précédents, pour lesquels on ne connaît pas d'algorithme permettant de repérer un optimum global (c'est-à-dire la meilleure solution possible), à coup sûr, et en un nombre fini de calculs. Les difficultés majeures, pour ce type de problème, sont les corrélations non identifiées entre les variables, le caractère bruité des fonctions ou encore des fonctions objectifs qui ne sont accessibles que par un dispositif expérimental. Il est à noter qu'il existe des problèmes à variables mixtes, c'est-à-dire que le problème présente à la fois des variables discrètes et continues.

Dans certaines situations, on peut se contenter de solutions de bonne qualité, sans garantie d'optimalité, mais au profit d'un temps de calcul réduit. Les méthodes approchées visent à générer des solutions de bonne qualité, en un temps de calcul "raisonnable", mais elles n'offrent pas de garantie pour trouver la solution optimale. Ces méthodes sont classées en deux sous-catégories : les algorithmes d'approximation et les heuristiques. Contrairement aux heuristiques, les algorithmes d'approximation permettent de quantifier la qualité de la solution trouvée par rapport à l'optimum. Enfin, il existe encore deux sous-classes de méthodes heuristiques : les heuristiques adaptées au problème considéré, et les métaheuristiques, qui regroupent des méthodes plus génériques.

2.3 Métaheuristiques

Les métaheuristiques constituent une classe d'algorithme stochastiques permettant de résoudre des problèmes d'optimisation. Elles ont en commun [Crep 08], en outre, les caractéristiques suivantes :

- Elles sont facilement adaptables à de nombreux problèmes sans changements majeurs dans leurs algorithmes, d'où le qualificatif "méta".
- Elles ont une capacité à résoudre un problème d'optimisation, à partir d'un nombre minimal d'informations.
- Elles n'offrent aucune garantie d'optimalité de la meilleure solution trouvée. Seule une approximation de l'optimum global est donnée. Ce constat n'est pas forcément un problème, du point de vue de la recherche opérationnelle, puisqu'il est souvent préférable d'avoir une approximation de l'optimum global trouvé rapidement qu'une valeur exacte trouvée en un temps rédhibitoire.
- Généralement d'origine discrète, elles ont l'avantage décisif, dans le cas continu, d'être discrètes, c'est-à-dire qu'elles ne recourent pas au calcul, souvent problématique, des gradients de la fonction objectif.
- Elles sont inspirées par des *analogies* : avec la physique (recuit simulé), avec la biologie (algorithmes génétiques) ou avec l'éthologie (essaim particulaire).
- Elles partagent les mêmes inconvénients : les difficultés de réglage des paramètres et le temps de calcul élevé.
- Elles ne s'excluent pas mutuellement : il est le plus souvent impossible de prévoir, avec certitude, l'efficacité d'une méthode donnée quand elle est appliquée à un problème donné. En effet, d'après les *No Free Lunch Theorems* [DWol 96], il n'existe pas d'algorithme d'optimisation qui, pour un problème donné, sera meilleur que tous les autres sur toutes les instances possibles. Dans la pratique, il existera toujours des instances pour lesquelles un algorithme sera mis en défaut par rapport à un autre.

La variété des métaheuristiques et leur capacité à résoudre des problèmes très divers ont contribué à leurs extensions. On trouve par exemple :

- Les métaheuristiques pour l'optimisation multimodale [Gold 87]. Il s'agit principalement de repérer tout un jeu d'optima globaux ou locaux.
- Les métaheuristiques pour l'optimisation multiobjectif ([Coll 02], [Shar 10a], [Deb 08], [Tiwa 09]). L'objectif n'est pas de trouver une solution unique, mais un ensemble de solutions non dominées, connu comme l'ensemble des solutions Pareto optimales.
- Les métaheuristiques pour l'optimisation dynamique ([Bran 02], [Our 02], [G Di 98]). Il

Chapitre I. État de l'art sur les métaheuristiques

s'agit de s'approcher de l'optimum à chaque pas de temps en faisant face à des variations temporelles de la fonction objectif.

- Les métaheuristiques hybrides ([Idou 02b], [Talb 02]). Le but est de trouver la meilleure combinaison de deux ou plusieurs métaheuristiques, afin d'en tirer les avantages respectifs.
- Les métaheuristiques parallèles ([Shar 10b], [Alba 05]). Le but visé par cette catégorie d'algorithmes est d'accélérer le calcul, en distribuant la charge de calcul sur plusieurs unités de calcul.

Chaque année, pour améliorer la résolution des problèmes les plus complexes, de nombreuses métaheuristiques sont proposées. De ce fait, un grand nombre de classes de métaheuristiques existe actuellement. La figure I.1 propose une classification générale des méthodes d'optimisation mono-objectif [Coll 02]. Parmi ces métaheuristiques, on trouve les métaheuristiques de *voisinage*, qui font progresser une seule solution à la fois (recuit simulé, méthode tabou) et les métaheuristiques distribuées, qui manipulent en parallèle toute une population de solutions (algorithmes génétiques, optimisation par essaim particulaire).

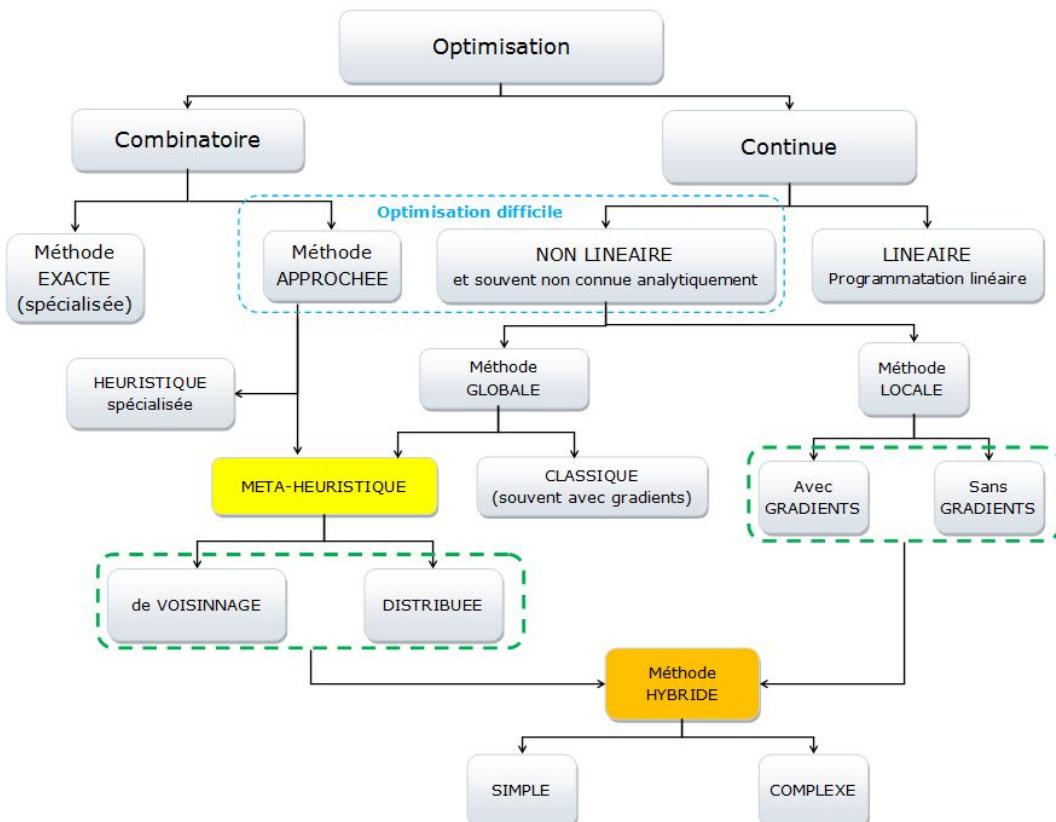


Figure I.1 – Classification générale des méthodes d'optimisation mono-objectif.

Algorithme 1: Un algorithme générique du recuit simulé.

```

1 Générer une solution initiale  $s_0$  ;  $s := s_0$ 
2  $s^* := s$  ;  $f^* := f(s)$ 
3  $T := T_0$ 
4 Tant que Critère d'arrêt n'est pas vérifié faire
5    $nb\_moves := 0$ 
6   Pour  $i \leftarrow 1$  à iter Faire
7     Générer une solution  $s_n$  voisine de  $s_c$ 
8     Calculer  $\Delta = f(s_n) - f(s_c)$ 
9     Si  $CritMetropolis(\Delta, T)$  Alors
10    |    $s_c := s_n$ 
11    |    $nb\_moves := nb\_moves + 1$ 
12    |   Si  $f(s) < f(s^*)$  Alors
13    |     |    $s^* := s$ 
14    |     |    $f^* := f(s)$ 
15    |   Fin
16   | Fin
17 Fin
18  $acceptance\_rate := i/nb\_moves$ 
19  $T := T * coeff$ 
20 Fin
21 Retourner  $s^*$ 
```

La suite de ce chapitre est dédiée à la présentation de quelques unes de ces méthodes.

2.3.1 Recuit simulé

Le recuit simulé (*SA*³) est une métaheuristique inspirée d'un processus utilisé en métallurgie [Kirk 83]. Pour modifier l'état d'un matériau, on dispose d'un paramètre de commande : la température. La technique consiste à chauffer préalablement le matériau pour lui conférer une énergie élevée. Puis on refroidit lentement le matériau en marquant des paliers de température de durée suffisante. Si la descente en température est très rapide, il apparaît des défauts et on obtient une structure méta stable caractérisée par des minima locaux d'énergie. La stratégie de baisse contrôlée de la température permet d'obtenir un état solide cristallisé, qui est un état stable correspondant à un minimum absolu d'énergie. En pratique, la technique exploite l'algorithme de Metropolis [Metr 53] qui permet de décrire le comportement d'un système en "équilibre thermodynamique" à une certaine température T : partant d'une configuration donnée, on fait subir à la solution courante s_c une modification

3. SA : *Simulated Annealing*.

Chapitre I. État de l'art sur les métaheuristiques

élémentaire. Si cette nouvelle solution s_n a pour effet de minimiser la fonction objectif (ou énergie) du système, elle est acceptée; si elle provoque, au contraire, une augmentation $\Delta = f(s_n) - f(s_c)$ de la fonction objectif, elle peut être acceptée tout de même, mais avec une certaine probabilité, comme le montre l'équation I.2.

$$p = \begin{cases} 1 & \text{si } f(s_n) - f(s_c) < 0 \\ \exp\left(\frac{-(f(s_n) - f(s_c))}{T}\right) & \text{sinon} \end{cases} \quad (\text{I.2})$$

On itère ensuite ce procédé, en gardant la température constante, jusqu'à ce que l'équilibre thermodynamique soit atteint, concrètement au bout d'un nombre "suffisant" de modifications. On abaisse alors la température, avant d'effectuer une nouvelle série de transformations.

L'équation I.3 donne un procédé souvent utilisé pour réduire la température, où $i = 0, 1, \dots$ et $\gamma = 0.99$.

$$T_{i+1} = \gamma T_i \quad (\text{I.3})$$

Le rôle de la température est primordial : à haute température, $\exp\left(\frac{-\Delta}{T}\right)$ est voisin de 1, donc la plupart des mouvements sont acceptés ; à basse température, $\exp\left(\frac{-\Delta}{T}\right)$ est voisin de 0, donc la plupart des mouvements augmentant l'énergie sont refusés ; à température intermédiaire, l'algorithme autorise, de temps en temps, des transformations qui dégradent la fonction objectif : il laisse ainsi au système une chance de s'extraire d'un minimum local. Le principe du recuit simulé est décrit dans l'algorithme 1.

Avantages et inconvénients : La méthode du recuit simulé a des avantages, car des solutions de bonne qualité sont obtenues pour un nombre important de problèmes, le plus souvent de grande taille. En outre, c'est une méthode générale, et facile à implémenter. Elle offre également une grande souplesse d'emploi vis-à-vis des évolutions du problème (incorporation de nouvelles contraintes).

Les inconvénients du recuit simulé résident, d'une part dans les réglages du nombre important de paramètres (température initiale, taux de décroissance de la température, durée des paliers de température, critère d'arrêt du programme, etc.) ; de bons réglages relèvent, souvent, du savoir-faire de l'utilisateur. Cependant, il existe des études qui s'attachent à régler de manière optimale les paramètres de l'algorithme [Siar 94]. L'autre inconvénient majeur de cette méthode est le temps de calcul, qui peut devenir très important dans certaines applications. Pour pallier à ce problème, plusieurs méthodes de parallélisation des calculs sont décrites dans [Azen 92].

Des concepts additionnels relatifs au sujet du recuit simulé et à quelques unes de ses applications peuvent être trouvés dans [Al k 08], [Li 08], [Sydo 09], [Mats 08], [Laza 08], [Siar 07].

2.3.2 Méthode Tabou

La méthode tabou (TS^4) [Gend 03] a été formalisée en 1986 par Glover. Sa principale particularité tient dans la mise en œuvre de mécanismes inspirés de la mémoire humaine. Le principe de base de la méthode tabou est simple : elle fonctionne avec une seule solution courante à la fois (au départ, une solution aléatoire), qui est actualisée au cours d'itérations successives. À chaque itération, le mécanisme de passage d'une configuration s à la suivante t comporte deux étapes :

- on construit l'ensemble des voisins de s , c'est-à-dire l'ensemble des configurations accessibles en un seul mouvement (une modification apportée à la solution courante) élémentaire à partir de s , soit $N(s)$ l'ensemble de ces voisins.
- on évalue la fonction objectif f du problème dans chacune des configurations appartenant à $N(s)$. La configuration t , qui succède à s dans la suite de solutions construite par la méthode tabou, est la configuration de $N(s)$ en laquelle f prend la valeur minimale. Notons que cette configuration t est adoptée même si elle est moins bonne que s , *i.e.* si $f(t) > f(s)$: c'est grâce à cette particularité que la méthode tabou permet d'éviter d'être bloqué dans les minima locaux de f .

Telle qu'elle est décrite, la procédure précédente est inopérante, parce qu'il y a le risque important de retourner à une configuration déjà retenue lors d'une itération précédente, ce qui engendre un cycle. Pour éviter ce phénomène, on tient à jour puis on exploite, à chaque itération, une liste de mouvements interdits. Cette liste, dite “liste de tabous”, contient m mouvements ($t \rightarrow s$), qui sont les inverses des m derniers mouvements ($s \rightarrow t$) effectués. Le principe général de la méthode tabou est donné dans l'algorithme 2.

L'algorithme modélise ainsi une forme rudimentaire de mémoire, la mémoire à court terme des solutions visitées récemment. Deux mécanismes supplémentaires, nommés intensification et diversification, sont souvent mis en œuvre pour doter aussi l'algorithme d'une mémoire à long terme. Ces processus n'exploitent plus la proximité dans le temps d'événements particuliers, mais plutôt la fréquence de leur occurrence, sur une période plus longue. L'intensification a pour but d'approfondir l'exploration de certaines régions de l'espace des solutions identifiées comme particulièrement prometteuses. La diversification est, au contraire, la réorientation périodique

4. TS : *Tabu Search*.

Chapitre I. État de l'art sur les métaheuristiques

Algorithme 2: Un algorithme générique de la méthode tabou

```
1 Générer une solution initiale  $s_0$  ;  $s := s_0$ 
2  $s^* := s$  ;  $f^* := f(s)$ 
3  $listTabu := \emptyset$ 
4 Tant que Condition d'arrêt n'est pas vérifiée faire
5    $mvt :=$  Meilleur mouvement parmi les mouvements non tabou ou exceptionnellement
      tabou (critère d'aspiration)
6    $s := s(+)mvt$ 
7   Si  $f(s) < f(s^*)$  Alors
8      $s^* := s$ 
9      $f^* := f(s)$ 
10  Fin
11  Mise à jour de  $listTabu$ 
12 Fin
13 Retourner  $s^*$ 
```

de la recherche d'un optimum vers des régions très rarement visitées jusqu'ici.

L'efficacité de la méthode tabou fait qu'elle est souvent employée pour résoudre des problèmes d'optimisation combinatoire : elle a été testée avec succès sur les grands problèmes classiques (ordonnancement, voyageur de commerce) et elle est fréquemment appliquée aux problèmes de routage, d'affectation quadratique, etc.

Avantages et inconvénients : La méthode tabou comporte moins de paramètres de réglage que le recuit simulé, ce qui la rend plus simple d'emploi. Cependant, les divers mécanismes annexes, comme l'intensification et la diversification, apportent une notable complexité. Sans oublier qu'il faudra trouver une valeur indicative (taille de la mémoire taboue) d'itérations pendant lesquelles les mouvements sont interdits. La gestion de cette mémoire peut devenir très lourde dans les cas où des stratégies de mémorisation complexe sont mises en œuvre.

2.3.3 Algorithmes génétiques

Les algorithmes génétiques (*GA*⁵) [Siva 07] sont inspirés de la théorie de l'évolution proposée par Charles Darwin et des processus biologiques qui permettent à des organismes de s'adapter à leur environnement :

- Dans chaque environnement, seules les espèces les mieux adaptées perdurent au cours du temps ; les autres étant condamnées à disparaître.

5. GA : *Genetic Algorithms*.

I.2 Problème d'optimisation mono-objectif

- Au sein de chaque espèce, le renouvellement des populations est essentiellement dû aux meilleurs individus de l'espèce.

On parlera ainsi d'individu dans une population et bien souvent l'individu sera résumé par un seul chromosome. Les chromosomes sont eux-mêmes constitués de gènes qui contiennent les caractères héréditaires de l'individu. On retrouvera aussi les principes fondamentaux de l'évolution naturelle, à savoir les principes de sélection, de croisement, de mutation, etc.

Dans le cadre de l'optimisation, chaque individu représente un point de l'espace d'état auquel on associe la valeur du critère à optimiser. On génère ensuite, aléatoirement, une population d'individus pour laquelle l'algorithme génétique s'attache à sélectionner les meilleurs individus, tout en assurant une exploration efficace de l'espace d'état.

Les algorithmes génétiques diffèrent des algorithmes classiques d'optimisation et de recherche essentiellement en quatre points fondamentaux :

- Ils utilisent un codage des éléments de l'espace de recherche et non pas les éléments eux-mêmes ;
- Ils recherchent une solution à partir d'une population de points et non pas à partir d'un seul point ;
- Ils n'imposent aucune régularité sur la fonction étudiée (continuité, dérivabilité, convexité, etc.). C'est un des gros atouts des algorithmes génétiques.
- Ils ne sont pas déterministes ; ils utilisent des règles de transition probabilistes.

L'algorithme génétique commence par générer une population d'individus, de façon aléatoire. Puis, on sélectionne un certain nombre d'individus dans la population, afin de générer une population intermédiaire, appelée aussi *mating pool*. Deux parents sont ensuite sélectionnés (P_1 et P_2) en fonction de leurs adaptations. On applique, aléatoirement, un opérateur de croisement avec une probabilité p_c qui génère deux enfants C_1 et C_2 . Si la probabilité de croisement p_c vaut 0.6, par exemple, on appliquera l'opérateur de croisement dans 60% des cas sur les individus P_1 et P_2 . On modifie ensuite certains gènes de C_1 et C_2 en appliquant un opérateur de mutation avec une probabilité p_m , ce qui produit deux nouveaux individus C'_1 et C'_2 pour lesquels on évalue le niveau d'adaptation avant de les insérer dans la nouvelle population. Contrairement à la reproduction et au croisement qui favorisent l'intensification, l'opérateur de mutation favorise la diversification des individus. On réitère les opérations de sélection, de croisement et de mutation afin de compléter la nouvelle population. Enfin, on réitère les opérations précédentes à partir de la sélection, jusqu'à ce qu'un critère d'arrêt soit satisfait. Différents critères d'arrêt de l'algorithme peuvent être choisis : nombre de générations fixé, limite de convergence de la population, population qui n'évolue plus suffisamment, etc. L'algorithme 3 résume le fonctionnement d'un algorithme génétique de base.

Chapitre I. État de l'art sur les métaheuristiques

Algorithm 3: Fonctionnement d'un algorithme génétique de base.

- 1 Générer aléatoirement une population de n individus.
 - 2 Evaluer la fitness $f(x)$ de chaque individu x de la population.
 - 3 **Tant que** Condition d'arrêt n'est pas vérifiée **faire**
 - 4 Sélectionner deux individus parents P_1 et P_2 dans la population.
 - 5 Avec une probabilité p_c , appliquer l'opérateur de croisement sur les deux parents P_1 et P_2 pour former deux nouveaux enfants C_1 et C_2 .
 - 6 Avec une probabilité p_m , appliquer l'opérateur de mutation à chacun des deux enfants C_1 et C_2 pour former deux nouveaux enfants C'_1 et C'_2 .
 - 7 Placer les deux enfants C'_1 et C'_2 dans la nouvelle population.
 - 8 **Fin**
 - 9 **Retourner** Meilleure individu
-

Il n'y a pas une seule manière de définir les mécanismes des algorithmes génétiques. Le codage et l'évaluation des individus, la sélection, le croisement et la mutation peuvent différer d'un problème à un autre (et chaque utilisateur aura même ses préférences). Pour une liste des applications des algorithmes génétiques, le lecteur intéressé peut consulter l'article [Zhen 99].

Avantages et inconvénients : D'abord, les algorithmes génétiques sont coûteux en temps de calcul, puisqu'ils manipulent plusieurs solutions simultanément. C'est le calcul de la fonction objectif qui est le plus pénalisant, et on optimise généralement l'algorithme de façon à éviter d'évaluer trop souvent cette fonction.

Ensuite, l'ajustement d'un algorithme génétique est délicat. L'un des problèmes les plus caractéristiques est celui de la dérive génétique, qui fait qu'un bon individu se met, en l'espace de quelques générations, à envahir toute la population. On parle dans ce cas de convergence prématuée, qui revient à lancer une recherche locale autour d'un minimum, qui n'est pas forcément l'optimum attendu. Les méthodes de sélection proportionnelle peuvent en particulier favoriser ce genre de dérive. Un autre problème surgit lorsque les différents individus se mettent à avoir des performances similaires : les bons éléments ne sont alors plus sélectionnés, et l'algorithme ne progresse plus.

Le choix d'une représentation "intelligente" pour permettre un remplacement générationnel efficace est un autre aspect de la question, et l'efficacité d'un algorithme génétique dépend beaucoup de la façon dont on opère le croisement des individus.

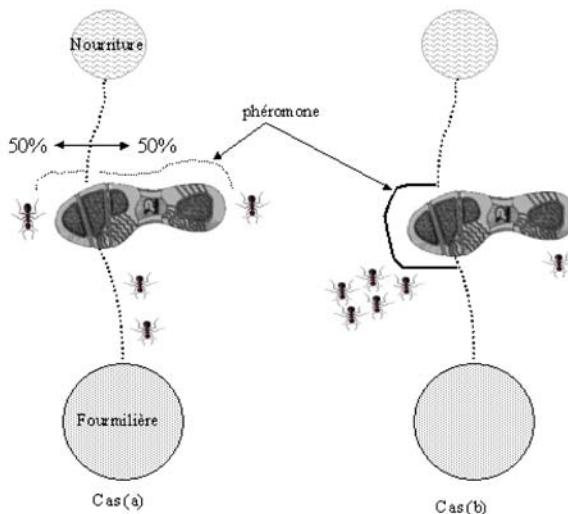
Le grand avantage des algorithmes génétiques est qu'ils parviennent à trouver de bonnes solutions sur des problèmes très complexes, et trop éloignés des problèmes combinatoires classiques pour que l'on puisse tirer profit de certaines propriétés connues. Ils doivent simplement

I.2 Problème d'optimisation mono-objectif

déterminer entre deux solutions quelle est la meilleure, afin d'opérer leurs sélections. On les emploie dans les domaines où un grand nombre de paramètres entrent en jeu, et où l'on a besoin d'obtenir de bonnes solutions en quelques itérations seulement - dans les systèmes de régulation de transport en temps réel, par exemple.

Par ailleurs, les algorithmes génétiques se prêtent bien, du fait de leur traitement simultané de solutions, à la recherche d'optimums multiples : en créant une fonction de coût partagée, dont la valeur dépend partiellement de la distance entre les individus, on voit se former graduellement des sous-populations d'individus, qui se stabilisent autour des différents pics de la fonction objectif. C'est la technique du "nichage" par la méthode du partage.

2.3.4 Colonie de fourmis



Cas(a) : à la rencontre d'un obstacle la fourmi décide de le contourner à droite ou à gauche avec le même pourcentage.
Cas(b) : après un certain temps (évaporation de phéromone) les chemins les plus courts qui mènent à la source de nourriture sont les plus marqués par les fortes trace de phéromone.

Figure I.2 – Colonie de fourmis.

L'idée d'imiter le comportement des fourmis pour trouver de bonnes solutions à des problèmes d'optimisation combinatoire a été proposée par Colorni, Dorigo et Maniezzo [Colo 91]. Le principe de cette méta-heuristique est basé sur la manière dont les fourmis cherchent leur nourriture et retrouvent leur chemin pour retourner dans la fourmilière. Initialement, les fourmis explorent les environs de leur nid de manière aléatoire. Sitôt qu'une source de nourriture est repérée par une fourmi, son intérêt est évalué (quantité, qualité) et la fourmi ramène un peu de nourriture au nid. Pour retrouver son chemin, elle a pris soin de laisser derrière elle une

Chapitre I. État de l'art sur les métaheuristiques

Algorithme 4: Principe de l'algorithme basé sur une colonie de fourmis.

- 1 Générer aléatoirement une population de n fourmis.
 - 2 Initialiser les traces.
 - 3 **Pour toute** fourmis et tant que le critère d'arrêt n'est pas satisfait **Faire**
 - 4 Construire une nouvelle solution à l'aide des informations contenues dans les traces et une fonction d'évaluation partielle.
 - 5 Évaluer la qualité de la solution.
 - 6 Mettre à jour les traces. - 7 **Fin**
 - 8 **Retourner** Meilleure solution
-

phéromones, trace chimique qu'elle arrive à détecter. Durant le chemin du retour, elle dépose également une quantité de phéromone dépendant de l'intérêt de la source de nourriture. Comme toutes les fourmis font de même, les traces laissées augmentent plus rapidement pour les sources de nourriture proches de la fourmilière, et lorsque plusieurs traces mènent à la même source, les traces correspondant aux chemins les plus courts sont renforcées à un rythme plus élevé (cas a dans la figure I.2). Il en résulte qu'après un certain temps, les chemins les plus rapides menant aux sources de nourriture les plus importantes sont marqués par de fortes traces (cas b).

Pour transposer ce comportement à un algorithme général d'optimisation combinatoire, on fait une analogie entre l'aire dans laquelle les fourmis cherchent de la nourriture et l'ensemble des solutions admissibles du problème, entre la quantité ou la qualité de la nourriture et la fonction-objectif à optimiser et enfin entre les traces et une mémoire adaptative. Une description détaillée de ces analogies pour le problème du voyageur de commerce peut être trouvée dans [Dori 97a, Dori 96]. Très schématiquement, un algorithme basé sur une colonie de fourmis peut être décrit par l'algorithme 4.

Une particularité de cet algorithme est qu'il a été pensé pour une exécution concurrente, contrairement à la recherche tabou ou le recuit simulé. Le point crucial de cette méta-heuristique est la définition des traces et leur mise à jour selon le problème à résoudre. Pour un voyageur de commerce par exemple, on peut prévoir une trace pour chaque paire de villes. Initialement, deux mécanismes d'utilisation des traces ont été imaginés : l'exploration et l'exploitation. L'exploration consiste à choisir la prochaine composante utilisée pour construire la solution (par exemple une arête dans le cas du voyageur de commerce) avec une probabilité proportionnelle à la valeur de la trace associée à cette composante. L'exploitation consiste à choisir la composante qui optimise une fonction mariant la valeur de la trace et la fonction d'évaluation partielle (par exemple, pour le voyageur de commerce, une combinaison linéaire de la valeur de la trace et de la longueur de l'arête correspondante). La mise à jour des traces a été proposée comme suit :

tout d'abord, la valeur de toutes les traces est diminuée, pour simuler le phénomène d'évaporation des phéromones. Ensuite, les traces correspondant aux composantes choisies dans la construction de la solution sont renforcées d'une valeur croissant avec la qualité de la solution. Naturellement, les algorithmes basés sur une colonie de fourmis ont aussi évolué. Pour les faire converger plus rapidement, on peut par exemple ne faire la mise à jour des traces que pour la meilleure solution produite par les fourmis à l'itération courante ou même durant toute la recherche. Toutes les implémentations récentes [Dori 97b, Gamb 97, Stut 99] améliorent la solution produite par chaque fourmi à l'aide d'une recherche locale gloutonne.

2.3.5 Optimisation par essaim particulaire

L'optimisation par essaim particulaire (*PSO*⁶) est une technique relativement récente développée par [Kenn 95] et inspirée à l'origine du monde du vivant. Elle s'appuie, notamment, sur un modèle permettant de simuler le déplacement d'un groupe d'oiseaux, d'un banc de poissons ou d'un essaim d'abeilles. Il s'agit d'une méthode faisant appel à une population d'agents, appelés ici *particules*. Cette méthode est fondée sur la collaboration des individus entre eux. Ici, l'efficacité est due à la collaboration plutôt qu'à la compétition. En effet, cette idée veut qu'un groupe d'individus, peu intelligents, possède une organisation globale complexe. Ainsi, grâce à des règles de déplacement très simples (dans l'espace des solutions), les particules peuvent converger progressivement vers un minimum global. Des règles très simples comme “rester relativement proche des autres individus”, “aller dans la même direction”, “à la même vitesse”, suffisent à maintenir la cohésion du groupe tout entier et à permettre des comportements collectifs complexes et adaptés.

La méthode met en jeu de larges groupes de particules sous forme de vecteurs se déplaçant dans l'espace de recherche. Chaque $i^{\text{ème}}$ particule est caractérisée par sa position x_i et par un vecteur de changement de position appelé *vitesse* v_i . La $i^{\text{ème}}$ particule est représentée comme $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$. Le cœur de la méthode consiste à choisir comment définir v_i . D'après la socio-psychologie, chaque individu est influencé par son comportement passé et par celui de ses voisins pour décider de son propre déplacement. Autrement dit, à chaque itération, chaque particule se déplace en fonction des deux meilleures valeurs suivantes :

- La meilleure position passée de la particule, appelée *pbest*.
- La meilleure position obtenue parmi n'importe quel voisin (voisin dans le réseau social et non, nécessairement, dans l'espace), appelée *gbest*. Lorsqu'une particule considère une par-

6. PSO : *Particle Swarm Optimization*.

Chapitre I. État de l'art sur les métaheuristiques

tie de la population comme ses voisins topologiques (voisins dans l'espace de recherche), la meilleure valeur est une meilleure position locale, appelée dans ce cas *lbest*.

À chaque itération, ces deux meilleures valeurs sont combinées afin d'ajuster la vitesse et calculer ainsi un nouveau mouvement pour la particule. *pbest* est considéré comme étant le composant cognitif (*cognition component*) relatif à l'*expérience individuelle* et *gbest* (ou *lbest*) est considéré comme étant le composant social (*social component*) relatif à la *communication sociale*. Notons qu'il semble être communément admis qu'un voisinage social *gbest* (un individu x_2 ayant, par exemple, pour voisins les individus x_1 et x_3 , quelles que soient les localisations spatiales de x_1 , x_2 , x_3) donne de meilleurs résultats qu'un voisinage spatial *lbest* (fonction de la proximité des individus dans l'espace de recherche). La figure I.3 schématise le déplacement d'une particule selon ces informations.

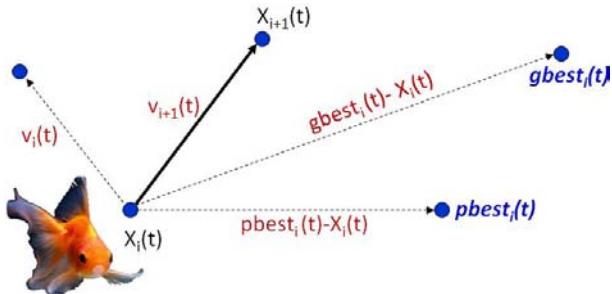


Figure I.3 – Mouvement de chaque particule.

Les équations de changement de position sont donc les suivantes :

$$v_{i+1} = \omega v_i + c_1 * \text{random}(0, 1) * (pbest_i - x_i) + c_2 * \text{random}(0, 1) * (gbest_i - x_i) \quad (\text{I.4})$$

$$x_{i+1} = x_i + v_{i+1} \quad (\text{I.5})$$

où *random*(0, 1) est un nombre aléatoire tiré dans l'intervalle [0, 1] et c_1 et c_2 sont deux paramètres qui ont pour rôle de pondérer les rôles relatifs de l'*expérience individuelle* (c_1) et de la *communication sociale* (c_2) (habituellement, on prend $c_1 = c_2 = 2$, voir [Shi 99]). Le tirage aléatoire uniforme est justifié si l'on ne considère aucun a priori sur l'importance de l'une ou l'autre source d'informations. Notons également qu'il est possible de considérer, à la place de

ce tirage aléatoire et uniforme, un tirage basé sur la distribution de Lévy, ou encore sur la distribution de Cauchy.

Pour éviter que le système n'explose en cas d'amplification d'oscillations trop grande, un paramètre V_{max} permet de limiter la vitesse (sur chaque dimension). Dans la version de base, c'est ce paramètre V_{max} qui va empêcher le système d'exploser par amplification des rétroactions positives. Dans le cas de l'équation I.4, si la somme à droite de l'égalité dépasse une certaine valeur constante, alors la vitesse sur cette dimension est initialisée à V_{imin} ou V_{imax} . Ainsi, les vitesses des particules sont limitées à l'intervalle $[V_{imin}, V_{imax}]$ qui sert comme contrainte de contrôle de la capacité d'exploration globale (la divergence) de l'algorithme. Cela réduit également la probabilité que les particules quittent l'espace de recherche. Notons que cela ne restreint pas les valeurs de x_i à l'intervalle $[V_{imin}, V_{imax}]$; mais limite seulement la distance maximale qu'une particule va parcourir au cours d'une itération. Cette méthode permet de provoquer une convergence de l'algorithme (l'amplitude des mouvements des individus diminue jusqu'à son annulation). On réalise ainsi un compromis efficace entre intensification et diversification.

Dans la même idée, une version met en place un poids d'inertie (*inertia weight*), noté ω dans l'équation de vitesse (voir équation I.4) [Shi 98]. Pour résumer, le poids d'inertie décroît en fonction du temps, ce qui provoque une convergence contrôlable par ce paramètre permettant ainsi de contrôler la dynamique intensification/diversification du système. La dynamique générale reste la même que dans la version précédente, à l'exception près de l'impossibilité de repartir dans une dynamique de diversification si un nouveau point, meilleur que le précédent, est trouvé.

Le lecteur intéressé trouvera un état de l'art complet entièrement dédié à l'optimisation par essaim particulaire et aux concepts qui lui sont associés dans [Cler 05].

2.4 Métaheuristiques hybrides

Les métaheuristiques hybrides exploitent la complémentarité de ces méthodes entre-elles, ainsi qu'avec d'autres approches "classiques". Cette nouvelle classe d'algorithmes a fait la preuve de sa robustesse et de son efficacité dans la résolution des problèmes d'optimisation difficiles. Nous proposons dans cette section quelques hybridations de la méthode d'optimisation par essaim particulaire (*PSO*) qui ont servi de base de comparaison pour nos algorithmes. D'autres approches hybrides sont décrites en détail dans [Dhae 05] et [Talb 02].

2.4.1 PSO hybride avec une recherche locale

Les dernières années, de nombreux auteurs ont proposé des méthodes hybrides combinant à la fois l'algorithme *PSO* et une méthode de recherche locale. L'algorithme *Tabu List PSO* (*TL-PSO*) décrit dans [Naka 08] est basé sur la combinaison de *PSO* et d'une méthode tabou (*TS*). Dans l'algorithme *PSO*, lorsqu'une particule trouve une solution optimale locale, toutes les autres particules se rassemblent autour d'elle et se retrouvent coincées dans cette solution locale. En revanche, la méthode *TS* peut sortir de ce minimum local et continuer à explorer l'espace de recherche.

La méthode *TL-PSO* essaie de combiner les avantages des deux méthodes *PSO* et *TS*. Pour ce faire, *TL-PSO* stocke l'historique des meilleures informations détenues par chaque particule (*pbest*) dans une liste tabou. Lorsque la capacité de recherche d'une particule se trouve réduite, la particule sélectionne un autre *pbest* à partir des valeurs de l'historique et l'utilise pour mettre à jour l'équation de vitesse. Ce mécanisme permet à chaque particule de rester active et la capacité de recherche de l'essaim progresse.

Dans [Wang 07], lors de l'exécution de *PSO*, si la meilleure valeur courante de la fonction objectif trouvée par l'essaim ne s'améliore plus depuis un certain nombre d'évaluations, deux actions sont entreprises. D'abord, les auteurs mettent la meilleure position courante dans une liste tabou et considèrent son voisinage comme étant tabou et ce afin d'éviter des attractions répétées vers cette zone dans les prochaines itérations. Ensuite, ils génèrent de façon répétitive des directions appropriées pour que la particule puisse s'extraire de ce minimum local, jusqu'à ce qu'elle se retrouve dans une zone non taboue.

Certains auteurs proposent des méthodes hybrides basées sur *PSO* et *SA*. Ainsi, dans [Prem 10], lorsqu'une particule du *PSO* stagne, *SA* est appliqué sur le résultat pour diversifier la position de la particule, même si la solution s'empire. D'autres travaux analogues peuvent être trouvés dans [Wang 04] et [Idou 09a].

2.4.2 PSO intégrant un opérateur génétique

Enfin, d'autres techniques hybrides se basent sur *PSO* et sur un des deux opérateurs génétiques (croisement ou mutation). Ainsi, dans [Pant 08], les auteurs ont évalué les performances de quatre algorithmes basés sur l'optimisation par essaim particulaire. Ces algorithmes sont : l'algorithme *PSO* standard, *Attraction-Repulsion based PSO* (*ATREPSO*), *Quadratic Interpolation based PSO* (*QIPSO*) et *Gaussian Mutation based PSO* (*GMPSO*). Tandis que tous les algorithmes présentés dans cet article sont guidés par la diversité de la population, afin de rechercher la solution globale optimale d'un problème d'optimisation donné, *QIPSO* utilise

I.2 Problème d'optimisation mono-objectif

l'opérateur de reproduction pour générer un nouveau membre de l'essaim et *GMPSO* utilise le concept de mutation.

En effet, l'algorithme *ATREPSO* est un algorithme à trois étapes, dans lequel les particules de l'essaim passent alternativement entre l'étape d'attraction (quand les particules sont attirées vers l'optimal global), l'étape de répulsion (les particules sont repoussées de la position optimale) et la phase intermédiaire (combinaison équilibrée d'attraction et de répulsion). Le déplacement des particules de l'essaim dans les différentes phases est contrôlé par la diversité de l'essaim.

L'algorithme *GMPSO* utilise les équations générales de position et de vitesse de la méthode *PSO* classique, auxquelles il ajoute un opérateur de mutation généré par une distribution gaussienne. Là encore, c'est la diversité de l'essaim qui décide quand appliquer la mutation (en dessous d'un seuil de diversité).

L'algorithme *QIPSO* utilise le concept de reproduction basé sur la “polygamie”. Il utilise la diversité de l'essaim comme mesure de guidage du croisement. Ainsi, lorsque la diversité passe en dessous d'un seuil minimal, l'opérateur de croisement quadratique non-linéaire est activé jusqu'à ce que la diversité atteigne un seuil maximal. Les seuils, minimal et maximal, de diversité sont prédéfinis par l'utilisateur.

L'algorithme *Gaussian PSO with jumps (GPSO-J)* proposé dans [Kroh 05] est fondé sur des sauts pour s'échapper des minima locaux. Lorsqu'il n'y a plus d'améliorations de la fonction *fitness* (minimum local), un compteur est incrémenté à chaque itération. Une fois que ce compteur atteint un seuil pré-spécifié, la particule effectue un saut vers un nouveau point. Cela est effectué grâce à l'introduction d'un opérateur de mutation qui change la position de la particule. L'opérateur de mutation est basé sur les distributions gaussienne et de Cauchy.

Dans l'algorithme *Particle Swarm Optimizer with Adaptive Tabu and Mutation (ATM-PSO)* décrit dans [Yu X 10], des mutations sont effectuées sous le guidage d'informations taboues de telle sorte que plusieurs mutations redondantes sont évitées. L'algorithme *ATM-PSO* s'exécute en deux étapes :

1. L'étape de détection tabou : durant l'exécution du *PSO* standard, pour une particule, s'il n'y a plus d'améliorations de sa fonction objectif durant quelques générations successives ; sa meilleure position est placée dans une liste tabou. Après quoi, toutes les particules proches de cette particule sont mutées, en utilisant une variable aléatoire de Cauchy.
2. L'étape de mutation et de contrôle : dans les générations suivantes, la position de chaque particule est contrôlée. Lorsqu'une particule se dirige vers sa région tabou, la mutation se produit. Un compteur tabou est attribué à chaque particule. Seules les particules ayant le compteur tabou le plus élevé seront mutées.

3 Problème d'optimisation multiobjectif

3.1 Introduction

La plupart des problèmes d'optimisation réels sont décrits à l'aide de plusieurs objectifs ou critères souvent contradictoires, devant être optimisés simultanément. Alors que, pour les problèmes n'incluant qu'un seul objectif, l'optimum cherché est clairement défini, celui-ci reste à formaliser pour les problèmes d'optimisation multiobjectif. En effet, pour un problème à deux objectifs contradictoires, la solution optimale cherchée est un ensemble de points correspondant aux meilleurs compromis possibles pour résoudre notre problème.

Les problèmes d'optimisation multiobjectif sont une généralisation à n fonctions objectif des problèmes d'optimisation classiques (cf. Section 1.1). Ils sont définis formellement comme suit :

$$\left\{ \begin{array}{l} \text{Optimiser} \quad \vec{F}(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})\} \\ \text{Tel que :} \quad C(\vec{x}) \leq 0 + \mathbb{R}^q \\ \text{Avec :} \quad \vec{x} \in \mathbb{R}^n, \quad \vec{C} \in \mathbb{R}^q \end{array} \right. \quad (\text{I.6})$$

Où m est le nombre d'objectifs ($m \geq 2$), $\vec{x} = (x_1, x_2, \dots, x_k)$ est le vecteur représentant les variables de décision, C représente l'ensemble des solutions réalisables associé à des contraintes d'égalité, d'inégalité et des bornes explicites (espace de décision), et \vec{F} est le vecteur à optimiser, c'est-à-dire à minimiser ou à maximiser. Sans perte de généralité, nous supposerons par la suite que nous considérons des problèmes de minimisation. Contrairement à l'optimisation mono-objectif, la solution d'un problème multiobjectif n'est pas unique, mais c'est un ensemble de solutions non dominées, connu comme l'ensemble des solutions Pareto Optimales (PO).

3.2 Méthodes de résolution

Dans la littérature, un grand nombre d'approches existent pour résoudre les problèmes multiobjectif ([Dhae 05], [Crepl 08], [Naki 07], [Jour 10], [Bari 03], [Meun 02], [Coor 08], [Talb 99]). Certaines approches utilisent des connaissances du problème pour déterminer des préférences sur les critères et ainsi contourner l'aspect multiobjectif du problème. Tandis que d'autres approches mettent tous les objectifs au même niveau d'importance, mais là aussi il existe plusieurs façons de réaliser une telle opération. Dans cette sous-section, on dresse une liste non exhaustive des travaux existant en optimisation multiobjectif. Le lecteur intéressé pourra consulter [Coll 02], [Deb 01], [Ehrg 00a], [Ehrg 00b], [Miet 99], [Ulun 94] pour compléter l'état de l'art sur les algorithmes multiobjectifs.

I.3 Problème d'optimisation multiobjectif

Toutes ces approches peuvent être classées en trois catégories :

- **Approche agrégative :**

Ce type d'approche a l'avantage d'être simple à mettre en œuvre et elle ne fournit qu'une seule solution, ce qui évite l'intervention d'un décideur. Le principe d'une telle approche est de transformer le problème multiobjectif en un problème mono-objectif pour y appliquer les méthodes de résolution déjà existantes [Coll 02]. La fonction objectif à optimiser est obtenue à l'aide d'une combinaison linéaire des différents objectifs f_i .

$$F(\vec{x}) = \sum_{i=1}^m w_i \times f_i(\vec{x}) \quad (\text{I.7})$$

où $w_i > 0$ correspond au poids affecté à l'objectif i ($\sum_{i=1}^m w_i = 1$).

Les principaux problèmes de ce type d'approche sont la détermination des paramètres w_i et le choix des interactions entre les différents objectifs. [Pars 02] propose d'agréger les objectifs suivant trois méthodes différentes : une agrégation linéaire classique, pour laquelle les poids sont fixés, une agrégation dynamique, pour laquelle les poids sont modifiés au cours du traitement et une agrégation dite “bang-bang” [Jin 01], pour laquelle les poids sont brutalement et aléatoirement [Ishi 98] modifiés au cours du temps.

D'un point de vue géométrique, l'agrégation peut être vue sous forme de projection du vecteur objectif \vec{f} sur le vecteur poids \vec{w} . Ainsi, le choix d'une direction de projection peut introduire une perte d'information, ce qui peut constituer une limite de l'approche. Parmi les approches agrégatives, on peut citer deux versions du recuit simulé en optimisation multiobjectif ([Frie 93], [Sera 94]). Les deux méthodes les plus populaires sont :

- la méthode “Multiple Objective Simulated Annealing” (MOSA) [Ulun 99] qui utilise les caractéristiques du recuit simulé pour rechercher le plus efficacement possible les solutions non-dominées.
- la méthode “Pareto Archived Simulated Annealing” (PASA) [Engr 97] qui utilise une fonction d'agrégation des fonctions objectifs, couplée avec un système d'archivage des solutions non-dominées.

Aussi, on peut citer la méthode d'optimisation par essaim particulaire qui a été proposée par [Baum 04] et qui utilise une topologie complètement connectée combinée avec une agrégation linéaire des objectifs. La population de l'essaim est divisée en k sous-essaims utilisant chacun un jeu de poids différent et possédant chacun leur propre “leader” ($gbest$). Les solutions optimales au sens de Pareto sont déterminées par le biais d'une méthode basée sur l'utilisation du gradient.

– Approche non-Pareto :

La caractéristique des approches non-Pareto est le traitement des objectifs séparément. Dans la littérature, on trouve deux types de méthodes :

- les méthodes à sélection lexicographique, qui tentent d'optimiser les objectifs séquentiellement, suivant un ordre prédéfini qui permet, par conséquent, de fixer le poids des objectifs. A titre d'exemple, [Hu 02] propose un algorithme *PSO*, avec une topologie circulaire de voisinage, qui optimise un seul objectif à la fois, en utilisant un ordre lexicographique. Cet algorithme n'introduit pas d'archive externe de stockage. Par contre, [Hu 03] propose une version de cet algorithme avec l'utilisation d'une archive externe. D'autres métaheuristiques ont été utilisées pour la résolution des problèmes multiobjectif à sélection lexicographique [Talb 99].
- les méthodes à sélection parallèle. [Scha 85] propose un algorithme nommé *VEGA* (*Vector Evaluated Genetic Algorithm*) qui sélectionne les solutions courantes du front de Pareto suivant chaque objectif indépendamment des autres. L'analyse du comportement de cet algorithme a montré qu'il se comportait d'une manière similaire à un algorithme utilisant une méthode agrégative. Dans [Pars 04], les auteurs ont proposé un algorithme parallèle basé sur l'approche *VEPSO* (*Vector Evaluated Particle Swarm*) qui est d'ailleurs inspirée de *VEGA*. Cette approche parallèle est basée sur un algorithme multi-essaim pour lequel chaque sous-essaim traite un des objectifs. Chaque sous-essaim échange son leader (*bbest*) avec les autres sous-essaims. Les auteurs montrent que cette méthode peut mener à des solutions Pareto optimales. Dans [Chow 04], un autre algorithme multi-essaim est proposé. Chaque essaim traite individuellement un des objectifs. L'échange des informations *gbest* est effectué entre les sous-essaims voisins. Chaque sous-essaim utilise une topologie basée sur le voisinage global pour son évolution et une topologie locale de voisinage pour communiquer avec les différents sous-essaims. D'autres travaux sur ce type de méthodes sont décrits dans [Rich 89] et [Berr 01].

– Approche Pareto :

Ce type d'approche utilise la notion de dominance dans son processus de recherche. En effet, résoudre un problème multiobjectif à l'aide d'une approche Pareto consiste à sélectionner les solutions non-dominées trouvées par l'algorithme au cours de son exécution. De ce fait, il apparaît nécessaire de tenir à jour une liste des solutions non-dominées trouvées au cours de l'exécution de l'algorithme et de la mettre à jour à la fin de chaque itération. Une archive extérieure est couramment utilisée pour résoudre ce problème. Chaque solution potentielle est ajoutée à l'archive si et seulement si elle n'est dominée par aucune des

I.3 Problème d'optimisation multiobjectif

solutions contenues dans l'archive. Par contre, si elle domine des solutions présentes dans l'archive, celles-ci seront alors retirées de l'archive. Le problème d'une telle approche est qu'elle peut être très rapidement coûteuse en complexité. En effet, une taille plus grande de l'archive entraîne une augmentation du temps de calcul nécessaire à la mise à jour des solutions non-dominées. Dans le pire des cas, si toutes les solutions d'une population sont ajoutées à l'archive, la complexité sera alors de l'ordre de $O(kNM^2)$, où k est le nombre d'objectifs, N est le nombre d'itérations et M est la taille de la population. Pour remédier à cet inconvénient, la plupart des approches Pareto fixent une taille maximale pour l'archive [Coel 02b]. Le choix de la solution finale revient donc au décideur, qui doit choisir, parmi l'ensemble fourni, la solution qui lui convient le mieux [Talb 99].

Parmi les algorithmes utilisant l'approche Pareto, on distingue deux catégories :

- les algorithmes non-élitistes. Notamment, l'algorithme MOGA⁷ [Fons 95] qui se base sur le classement des individus suivant le nombre d'individus dominés. L'algorithme NSGA⁸ [Sriv 95] qui propose de diviser la population en plusieurs groupes, fonctions du degré de domination de chaque individu. Enfin, l'algorithme qui fait référence dans le domaine est NSGA-II [Deb 02], qui est une évolution de NSGA, mais en version élitiste.
- les algorithmes élitistes. On peut citer, par exemple, l'algorithme SPEA⁹ [Zitz 99], où l'on profite du passage de génération pour mettre à jour les différentes sauvegardes. Les individus non-dominés sont sauvegardés et les individus dominés déjà présents sont éliminés. Une évolution de cet algorithme est disponible dans [Zitz 01]. Cette évolution a une plus grande complexité, mais présente de meilleurs résultats.

Pour les approches hybrides, de nombreux travaux tentent d'hybrider les algorithmes évolutionnaires et l'algorithme PSO. Par exemple, dans [Ray 53] les auteurs proposent un algorithme combinant l'algorithme *PSO* avec des opérateurs issus des algorithmes évolutionnaires, tout en utilisant la dominance au sens Pareto. L'algorithme ainsi proposé utilise par exemple un opérateur de densité sur le voisinage pour promouvoir la diversité au sein de l'essaim. Dans [Coel 02a] et [Coel 04], les auteurs proposent un algorithme basé sur l'utilisation d'une archive externe mise à jour suivant un critère géographique. L'espace de recherche est divisé en hyper-cubes auxquels on donne des “notes”, qui sont fonction du nombre de solutions non-dominées situées dans chacun d'eux. En plus d'un opérateur de mutation, cet algorithme intègre un opérateur de sélection par “roulette” basé

7. Multiple Objective Genetic Algorithm

8. Non Dominated Sorting Genetic Algorithm

9. Strength Pareto Evolutionary Algorithm

Chapitre I. État de l'art sur les métaheuristiques

sur les notes, et utilisé pour choisir les leaders dans les hyper-cubes. Une autre approche hybride, proposée par [Srin 03], utilise les opérateurs des algorithmes évolutionnaires pour améliorer l'efficacité de l'algorithme *PSO*. Par exemple, un opérateur de sélection est utilisé pour assurer la convergence vers des solutions non-dominées, sans utiliser d'archive externe.

Le lecteur trouvera dans [Coll 02] une très bonne présentation détaillée de ces trois classes d'approches de résolution de problèmes d'optimisation multiobjectif. De nombreuses autres méthodes sont référencées dans [Reye 06] et [Zitz 99].

4 Conclusion

Dans ce chapitre, nous avons présenté les problèmes d'optimisation mono-objectif et multiobjectif. Pour résoudre ces problèmes, nous avons présenté quelques algorithmes basés essentiellement sur les métaheuristiques. Ces algorithmes sont de nature stochastique et visent à résoudre un très grand nombre de problèmes, sans modification majeure de leur structure. Les métaheuristiques sont inspirées d'analogies avec des domaines aussi variés que la génétique, la physique ou encore l'éthologie. Elles ont rencontré un vrai succès grâce à leur modularité et à leur simplicité de mise en œuvre. En effet, quelle que soit la nature du problème (mono-objectif/multiobjectif, continu/discret), les métaheuristiques sont facilement adaptables et/ou hybridables dans le but d'obtenir les meilleurs résultats/performances possibles.

Les chapitres suivants présentent quelques problèmes d'optimisation que nous avons résolus en utilisant des métaheuristiques de voisinage et à base de population (distribuées). Un intérêt particulier sera porté à l'hybridation de métaheuristiques pour résoudre des problèmes mono-objectif et multiobjectif.

Chapitre II

Métaheuristiques pour la planification des réseaux de radiodiffusion

Les travaux décrits dans ce chapitre ont donné lieu à plusieurs publications : [[Idou 01b](#)], [[Idou 02f](#)], [[idou 01a](#)], [[Idou 01c](#)], [[Idou 02d](#)], [[Idou 02b](#)], [[Idou 02c](#)], [[Idou 02e](#)], [[Idou 03](#)], [[Idou 04](#)], [[Idou 06](#)], [[Idou 09c](#)].

Nos deux articles joints à ce mémoire sont :

- **L. Idoumghar** and R. Schott, “Two Distributed Algorithms for the Frequency Assignment Problem in the Field of RadioBroadcasting”. Journal of IEEE Transactions on Broadcasting, Vol. 55, Issue 2, Part 1, pp. 223-229, June 2009.
- **L. Idoumghar** and Ph. Debreux, “New Modeling Approach for the Frequency Assignment Problem in Broadcasting”. Journal of IEEE Transactions on Broadcasting, pp. 293-298, Vol. 48, N. 4, December 2002.

Le travail présenté dans ce chapitre résume une partie de notre collaboration avec la société *TéléDiffusion de France* depuis 2001. Ce travail se poursuit actuellement et jusqu'en 2015, puisque nous avons démarré le 1^{er} janvier 2012 la thèse CIFRE d'Akram Bedoui que je co-encadre à 50% avec le Professeur René SCHOTT.

1 Introduction

La planification d'un réseau de radiodiffusion, de type TV numérique ou radio FM, constitue l'une des tâches les plus complexes à la charge de l'opérateur du réseau. Elle est considérée comme un processus très délicat dont le résultat conditionne le succès des opérateurs. En effet, un réseau mal planifié se traduit par une qualité de service médiocre, en termes de zones de service ou de pourcentage de population desservie. Cela entraîne des dépenses supplémentaires, donc un manque à gagner pour l'opérateur. La planification a pour objectif de minimiser le coût

Chapitre II. Métaheuristiques pour la planification des réseaux de radiodiffusion

de l'infrastructure radio et réseau, tout en tenant compte des caractéristiques de l'environnement ou de la région à servir (caractéristiques géographiques et de propagation radio) et de la bande de fréquences à utiliser. Le processus de planification doit être capable de fournir le meilleur emplacement des émetteurs et leurs caractéristiques en terme de puissance. Il doit aussi produire un meilleur plan de fréquences, qui permet de spécifier, pour chaque émetteur, la meilleure fréquence à lui affecter afin de maximiser sa zone de service. Enfin, il doit produire un plan d'équipements et de connexion réseau entre toutes ces entités.

Le processus de planification peut être décomposé en deux grandes étapes :

1. La première étape, appelée *dimensionnement* ([Crep 05], [Crep 06], [Vasq 01], [Hurl 01]) de réseau, représente une étape très importante, car elle consiste à déterminer le nombre, l'emplacement des émetteurs et la configuration de leurs paramètres (type d'antenne, puissance, etc.). Un réseau radioélectrique comprend donc un ensemble d'émetteurs fixes qui desservent chacun une zone géographique appelée *zone de couverture*. L'ensemble de ces zones vise à assurer une couverture optimale ou quasi-optimale d'une zone géographique à couvrir.
2. La deuxième étape du processus de planification des réseaux radioélectriques, appelée *allocation de fréquences* ([Coup 99], [Crom 94], [Dorn 95], [Tiou 95]) a pour but d'allouer une ou plusieurs fréquences pour chaque point de service (émetteur) e du réseau, tout en minimisant les interférences dues aux fréquences des émetteurs brouilleurs de e . Il s'agit d'une application complexe dans le domaine des télécommunications. La résolution de ce problème nécessite la mise en place d'une modélisation spécifique, qui permet de mieux prendre en compte les spécificités du réseau de chaque opérateur. Ces différentes modélisations donnent naissance à différents problèmes d'allocation de fréquences.

Dans la littérature, il existe deux méthodes d'allocation de fréquences. La première est de nature *statique* [Coup 99] : les fréquences affectées/allouées aux émetteurs le sont de manière permanente (cas des systèmes de radiodiffusion notamment). La seconde méthode est de nature *dynamique* ([Crom 94], [Dorn 95], [Tiou 95]) : aucune fréquence n'est attribuée à un émetteur de manière permanente. Donc une fréquence peut être utilisée dans n'importe quelle région à condition de satisfaire les contraintes d'interférence. Dans cette dernière méthode, utilisée plus particulièrement dans le cadre des radiocommunications ([Sing 97], [Zhan 89]), une fréquence peut servir un appel dans n'importe quelle cellule, si les contraintes ne sont pas violées.

Dans ce chapitre, nous nous intéressons à l'aspect statique du problème d'allocation de fréquences dans le cadre de la radiodiffusion. Cet intérêt est justifié par l'affectation

permanente des fréquences, aux diffuseurs, par les autorités de régulation. Par conséquent, aucun changement dynamique de fréquence n'est autorisé.

2 Modélisation par contraintes

2.1 Description

Le problème d'allocation de fréquences en radiodiffusion peut être modélisé sous la forme du problème d'optimisation suivant : *Soit $G(E, V)$ un graphe pondéré tel que V correspond à la liste des émetteurs du réseau à planifier et E l'ensemble d'arêtes pondérées correspondant aux différentes contraintes d'interférence à satisfaire. Le problème est de trouver une allocation qui satisfasse les différentes contraintes et qui minimise les niveaux d'interférence. Idéalement, on cherche à attribuer aux émetteurs différentes fréquences non interférantes ou à s'y rapprocher au maximum en respectant les contraintes d'interférence.*

Pour résoudre ce problème, et avant la mise en œuvre d'une méthode d'optimisation, nous devons disposer d'une modélisation des contraintes d'interférence. Cette modélisation, communément utilisée, se base sur le principe de symétrie de contraintes entre deux émetteurs qui interfèrent. Nous allons démontrer, lors des simulations, que ce principe n'est pas suffisant.

La figure II.1 illustre un exemple de graphe de contraintes.

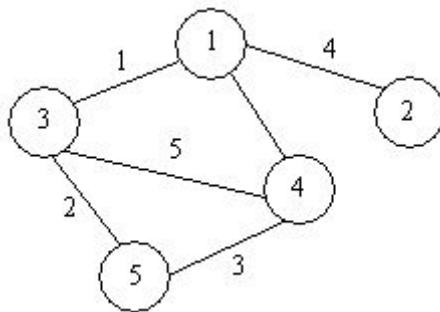


Figure II.1 – Exemple de graphe de contraintes.

Pour résoudre ce problème, *NP-Difficile*, nous présentons dans ce chapitre deux des mét-heuristiques associées aux deux modélisations que nous avons proposées.

Remarque 1

En radiodiffusion, les poids des arêtes (écart en canaux) peuvent prendre des valeurs dans $\{1,2,3,4,5\}$ contrairement au domaine de la radiocommunication, où les contraintes sont soit 1 soit 2. Une autre différence entre la radiodiffusion et la radiocommunication est que les émetteurs

de la radiodiffusion ont une puissance plus élevée, ce qui entraîne que le nombre de brouilleurs potentiels d'un émetteur en radiodiffusion est largement supérieur à six (qui est souvent le cas en radiocommunication).

2.2 Algorithme génétique hybride

La combinatoire élevée du problème d'allocation de fréquences nous a conduits à nous intéresser à une classe d'algorithmes d'optimisation stochastique fondée sur les mécanismes de la sélection naturelle et de la génétique. Il s'agit des algorithmes génétiques (AGs) [Gold 89] (voir 2.3.4)

Notre intérêt pour ces algorithmes est justifié par la simplicité de leurs mécanismes, la facilité de leur mise en application et leur efficacité, même pour des problèmes complexes, bien qu'ils souffrent d'un phénomène appelé “convergence prématuée” vers un minimum local. Ce phénomène apparaît suite aux mauvais paramétrages de l'algorithme, ou à la non-adéquation des opérateurs génétiques (mutation, croisement, sélection) au problème étudié.

Pour pallier à cet inconvénient, nous avons proposé, pour résoudre le problème d'allocation de fréquences, un algorithme génétique hybride, appelé *HGA*, qui est détaillé dans [Idou 02b] pour sa version séquentielle. La version distribuée est détaillée dans [Idou 09c].

HGA, décrit dans l'algorithme 5 (resp. 6) pour la version séquentielle (resp. distribuée), se base sur le principe suivant : les opérateurs évolutionnaires doivent favoriser la découverte de régions d'exploration plus intéressantes, tandis que l'opérateur de mutation génère des solutions dans un espace de recherche restreint. L'opérateur de mutation utilisé par *HGA* est basé sur une variante de la méthode tabou [Glov 97]. Cette variante, que nous avons baptisée “Méthode tabou probabilisée” (*Probabilistic Tabu Search*), est décrite dans l'algorithme 7 et a pour but d'affiner la recherche dans un espace restreint.

L'algorithme 5 est basé sur le principe de *steady state*, dans lequel un sous-ensemble de la population est changé à chaque itération. Dans cet algorithme, *PopSize* représente la taille de la population P_i , *maxGen* correspond au nombre maximum de générations à réaliser et p_c (resp. p_m) correspond à la probabilité de croisement (resp. mutation).

Pour mettre en place cet algorithme, nous avons défini les éléments suivants :

1. **Codage** : nous avons choisi de représenter nos solutions sous la forme d'un tableau dont les indices correspondent aux numéros des émetteurs et le contenu aux fréquences attribuées à ces émetteurs. Dans le cas où l'on aurait un scénario de replanification, les émetteurs non replanifiables¹ seront stockés dans un unique tableau, qui sera commun

1. Émetteurs possédant des fréquences fixes.

Algorithme 5: Version séquentielle de l'algorithme génétique hybride (*HGA*)

```

1 Construire le graphe des contraintes
2 Générer une sous-population  $P_0$ 
3 Evaluer  $P_0$  et trouver la meilleure solution  $\pi^*$ 
4  $\pi_{Elite} \leftarrow \pi^*$ ,  $i \leftarrow 1$ 
5 Tant que Critère d'arrêt n'est pas vérifié Faire
6    $P_i \leftarrow \emptyset$ 
7   Pour  $j \leftarrow 1$  à  $PopSize/2$  Faire
8     Sélectionner deux parents  $I_1$  et  $I_2$  dans  $P_{i-1}$ , Enfants  $\leftarrow (I_1, I_2)$ 
9     Avec une probabilité  $p_c$ , exécuter Enfants := croisement( $I_1, I_2$ )
10    Avec une probabilité  $p_m$ , muter les Enfants en utilisant la méthode Tabou Probabilisée
11    Evaluer les Enfants et les ajouter à  $P_i$ 
12  Fin
13  Ajouter  $P_{i-1}$  à  $P_i$ 
14  Trier  $P_i$ 
15  Garder  $PopSize$  meilleurs individus dans  $P_i$ 
16  Trouver le meilleur individu  $\pi^*$  dans  $P_i$ 
17  Si  $\pi_{Elite} = \pi^*$  Alors
18    |  $cpt \leftarrow cpt + 1$ 
19  Fin
20  Si  $\pi^*$  est meilleur que  $\pi_{Elite}$  Alors
21    |  $cpt \leftarrow 0$  et  $\pi_{Elite} \leftarrow \pi^*$ 
22  Fin
23  Si  $fitness(\pi_{Elite}) = 0$  ou  $cpt = K$  ou  $i = maxGen$  Alors
24    | terminer HGA
25  Fin
26   $i \leftarrow i + 1$ 
27 Fin

```

à toutes les solutions (individus). Par conséquent, un individu est réduit à un tableau contenant uniquement les émetteurs replanifiables. Pour avoir la solution réelle, il suffit de concaténer les deux tableaux.

2. **Population initiale :** chaque algorithme génétique nécessite une population initiale P_0 qui sert de point de départ. Cette population est générée de manière aléatoire.
3. **Fonction objectif :** la fonction objectif permet de mesurer la qualité de chaque individu. La fonction que nous avons proposée est définie ainsi :

$$F = \sum_{i,j} \rho_{i,j} \delta_{i,j}$$

avec $(i, j) \in$ ensemble des arêtes du graphe des contraintes G , $\rho_{i,j}$ représente le poids

Chapitre II. Métaheuristiques pour la planification des réseaux de radiodiffusion

Algorithme 6: Version distribuée de l'algorithme *HGA* s'exécutant sur chaque processeur

```

1 Construire le graphe des contraintes
2 Générer une sous-population  $P_0$ 
3 Evaluer  $P_0$  et trouver la meilleure solution  $\pi^*$ 
4  $\pi_{Elite} \leftarrow \pi^*$ ,  $i \leftarrow 1$ 
5 Tant que Critère d'arrêt n'est pas vérifié faire
6    $P_i \leftarrow \emptyset$ 
7   Pour  $j \leftarrow 1$  à  $PopSize/2$  Faire
8     Sélectionner deux parents  $I_1$  et  $I_2$  dans  $P_{i-1}$ , Enfants  $\leftarrow (I_1, I_2)$ 
9     Avec une probabilité  $p_c$ , exécuter Enfants := croisement( $I_1, I_2$ )
10    Avec une probabilité  $p_m$ , muter les Enfants en utilisant la méthode Tabou Probabilisée
11    Evaluer les Enfants et les ajouter à  $P_i$ 
12  Fin
13  Ajouter  $P_{i-1}$  à  $P_i$ 
14  Si Condition de migration est vérifiée Alors
15    | Réception des  $n$  individus et les ajouter à  $P_i$ 
16  Fin
17  Trier  $P_i$ 
18  Garder  $PopSize$  meilleurs individus dans  $P_i$ 
19  Trouver le meilleur individu  $\pi^*$  dans  $P_i$ 
20  Si  $\pi_{Elite} = \pi^*$  Alors
21    |  $cpt \leftarrow cpt + 1$ 
22  Fin
23  Si  $\pi^*$  est meilleur que  $\pi_{Elite}$  Alors
24    |  $cpt \leftarrow 0$  et  $\pi_{Elite} \leftarrow \pi^*$ 
25  Fin
26  Si  $fitness(\pi_{Elite}) = 0$  ou  $cpt = K$  ou  $i = maxGen$  Alors
27    | terminer HGA
28  Fin
29   $i \leftarrow i + 1$ 
30 Fin
```

associé à la contrainte $m_{i,j}$ ($\rho_{i,j}$ et $m_{i,j}$ sont des entiers positifs). $\delta_{i,j}$ est défini par :

$$\delta_{i,j} = \begin{cases} 0 & \text{Si } |f(v_i) - f(v_j)| \geq m_{i,j} \\ m_{i,j} - |f(v_i) - f(v_j)| & \text{Sinon} \end{cases}$$

où $f(v_i)$ représente la fréquence affectée à l'émetteur v_i .

La fonction objectif peut être utilisée pour mesurer la qualité de la fréquence attribuée à

chaque gène (émetteur v_i). Elle sera donc définie comme suit :

$$F_i = \sum_{j \in \text{adjacent_de}(i)} \rho_{i,j} \delta_{i,j} \quad (3)$$

Dans la suite de ce chapitre, nous appellerons cette fonction la *fonction de coût local*.

4. **Croisement** : Nous avons utilisé un opérateur de croisement uniforme [Sysw 89]. Cet opérateur a pour objectif de faire des sauts dans l'espace des solutions, tandis que l'opérateur de mutation, basé sur la méthode tabou probabilisée, permet d'affiner la recherche. Ce principe de fonctionnement est le contraire de ce qui est fait dans les AGs classiques, puisqu'ils utilisent l'opérateur de croisement pour tenter de créer de bons individus en manipulant la structure des chromosomes des parents ([Qi 93], [Sysw 89]) et c'est l'opérateur de mutation qui permet de faire des sauts dans l'espace des solutions.
5. **Mutation** : la mutation a pour but d'enrichir la diversité de la population. La version de base de la mutation, dite “mutation simple”, consiste à modifier aléatoirement, avec une probabilité p_m faible, les valeurs des gènes des chromosomes. Il est clair que cet opérateur ne permet pas de créer de meilleurs individus et son unique avantage est d'éviter l'établissement de populations uniformes incapables d'évoluer [Holl 92].

À l'inverse de ce principe, nous avons mis en place un nouvel opérateur de mutation basé sur la *méthode tabou probabilisée*. Cette méthode a pour but d'affiner la recherche d'un meilleur individu. Cet opérateur a donc pour objectif de produire de bons individus. L'originalité de cette méthode vient du fait qu'elle utilise un critère d'aspiration probabilisé. Notre idée de base était d'introduire des probabilités dans la méthode tabou classique (voir l'algorithme 7) et d'observer le comportement de cette méthode, en faisant la comparaison avec la meilleure solution obtenue par la méthode tabou implémentée par TDF-C2R et qui a fait l'objet d'un brevet [Coup 99]. Dans [Idou 02a], nous avons trouvé que la probabilité $p_t = 0,2$ permet d'obtenir le meilleur gain, en utilisant une instance de problème d'allocation de fréquences lié au *Réseau1* (plus de 5790 émetteurs et plus de 153480 contraintes).

2.3 Résultats et simulations

Dans le but de valider les résultats obtenus avec notre approche, nous avons utilisé des jeux de tests correspondant à des instances réelles du problème d'allocation de fréquences, dans le cadre de la radio FM en France. Dans ce chapitre nous présentons les résultats obtenus en utilisant quatre instances fournies par TéléDiffusion de France :

Algorithme 7: Méthode Tabou Probabilisée

```

1 Soient  $X$  une solution initiale et  $X^* \leftarrow X$  la meilleure solution
2  $\text{list\_tabu} \leftarrow \emptyset$ 
3 Tant que condition d'arrêt n'est pas vérifiée faire
4   Soit  $s^*$  la meilleure configuration, voisine de  $X$ , qui n'est pas interdite par la
      list_tabu. Cette configuration est choisie à l'aide de la probabilité  $p_t$ .
5   Choisir  $s_t$  la meilleure configuration, voisine de  $X$ , interdite par la list_tabu (Critère
      d'aspiration).
6   Choisir la meilleure configuration  $X$ , entre  $s_t$  et  $s^*$ , qui minimise la fonction objective.
7   Mettre à jour la list_tabu.
8   Mettre à jour la matrice des interférences  $M$ .
9   Si  $X$  est meilleure que  $X^*$  Alors
10    |  $X^* \leftarrow X$ 
11   Fin
12   Mettre à jour la condition d'arrêt
13 Fin

```

- instance 1 : nommée *Réseau1*, contient plus de 5790 émetteurs et plus de 153480 contraintes.
Ce réseau correspond à un exercice de replanification de 546 émetteurs, parmi les émetteurs qui constituent *Réseau1*.
- instance 2 : nommée *Réseau2*, contient plus de 970 émetteurs et plus de 43800 contraintes.
Ce réseau correspond à un exercice de replanification de 165 émetteurs, parmi les émetteurs qui constituent *Réseau2*.
- instance 3 : nommée *Réseau3*, contient plus de 970 émetteurs et plus de 25550 contraintes.
Ce réseau correspond à un exercice de replanification de 86 émetteurs, parmi les émetteurs qui constituent *Réseau3*.
- instance 4 : nommée *Réseau4*, contient plus de 970 émetteurs et plus de 12900 contraintes.
Ce réseau correspond à un exercice de replanification de 37 émetteurs, parmi les émetteurs qui constituent *Réseau4*.

Chaque émetteur peut prendre une fréquence parmi les 200 fréquences de la bande *FM*. En réalité, la bande *FM*, qui va de $87.6MHz$ à $107.9MHz$ en condition opérationnelle, est constituée de 204 canaux (avec un pas de $100kHz$).

2.3.1 Comparaison en terme de contraintes violées

Dans cette validation, il s'agit de comparer les différentes méthodes, selon le nombre et le type de contraintes violées résiduelles. Le procédé d'élimination des méthodes se base sur ces deux critères. Par exemple si, pour deux méthodes, on a le même nombre de contraintes

II.2 Modélisation par contraintes

violées, la méthode retenue est celle qui viole le moins de contraintes de degré le plus élevé. Ceci peut être justifié par le fait qu'une contrainte de degré plus élevé (5 par exemple) entre deux émetteurs se traduit par le fait que ces deux émetteurs peuvent se situer sur un même pylône (ou même position géographique) et possèdent les mêmes propriétés.

Le tableau II.1 donne les résultats, en moyenne, sur cinquante exécutions, obtenus en utilisant :

- *TS* : la méthode tabou décrite dans [Coup 99]
- *PTS* : la méthode tabou probabilisée utilisant le critère d'aspiration avec une probabilité égale à $p_t = 0,2$;
- *HGA* : notre algorithme génétique hybride.
- *Ant* : l'algorithme basé sur la colonie de fourmis [Come 98]

	<i>TS</i>	<i>PTS</i>	<i>Ant</i>	<i>HGA</i> _{séquentiel}
Nb. total de contraintes violées	460,80	444,04	466,5	366,66
nb. contraintes 1	337,32	319,44	302,2	261,42
nb. contraintes 2	120,44	121,6	159,1	102,26
nb. contraintes 3	2,94	2,84	5,2	2,98
nb. contraintes 4	0,14	0,1	0	0
nb. contraintes 5	0	0,06	0	0

Tableau II.1 – Comparaison des résultats, en moyenne, sur une cinquantaine d'exécutions, obtenus en utilisant quatre méthodes d'optimisation pour résoudre l'instance *Réseau 1*.

L'analyse des résultats donnés dans le tableau II.1 montre que *HGA* obtient de meilleurs résultats en comparaison des autres algorithmes. A titre d'exemple, *HGA* satisfait 20,43% de contraintes en plus par rapport à la meilleure solution obtenue par la méthode tabou [Coup 99].

Les tableaux II.2, II.3 et II.4 donnent les résultats de comparaison des deux versions d'*HGA* avec le plan de fréquence opérationnel en France pour chacune des instances (*Réseau2*, *Réseau3* et *Réseau4*).

L'analyse de ces trois tableaux montre qu'en comparaison avec le plan opérationnel en France, nos approches satisfont :

- 54,93% de contraintes en plus pour *Réseau2* ;
- 71,59 % de contraintes en plus pour *Réseau3* ;
- 77,68% de contraintes en plus pour *Réseau4*.

Remarque 2

Les versions séquentielle et distribuée de *HGA* sont implémentées en utilisant la bibliothèque MPICH2 (version 1.0.6). Pour la version distribuée, un *cluster* de 5 machines, tournant chacune

Chapitre II. Métaheuristiques pour la planification des réseaux de radiodiffusion

sous Windows XP Professional version 2002, est utilisé. Chaque machine est équipée d'un processeur Pentium (D) à 3 GHz et possède 1 Go de mémoire.

	HGA _{séquentiel}	HGA _{distribué}	ANT	Plan opérationnel
Nb. total contraintes violées	64	64	70	142
Nb. contraintes 1	20	23	17	37
Nb. contraintes 2	39	36	41	68
Nb. contraintes 3	4	4	9	15
Nb. contraintes 4	1	1	3	6
Nb. contraintes 5	0	0	0	16
Temps d'exécution	17 sec	6 sec	-	-

Tableau II.2 – Comparaison des résultats obtenus en résolvant l'instance correspondant au *Réseau2*.

2.3.2 Validation métier des plans de fréquences

Dans cette section, nous présentons la validation métier (sur le terrain) de nos solutions. Lors de cette validation, nous avons utilisé un outil, développé par TDF, qui prend en compte les brouillages multiples pour le calcul des zones de services et qui fournit des simulations proches de la situation réelle. Cet outil permet de calculer le pourcentage $\frac{\text{zone de service}}{\text{zone de couverture}}$ pour chaque émetteur.

Rappelons que la zone de couverture d'un émetteur e est définie comme la zone couverte par cet émetteur en l'absence de tout brouilleur. La zone de service de e est la zone couverte par cet émetteur en présence de tous les brouilleurs. La zone de service est par conséquent toujours plus restreinte que la zone de couverture.

	HGA _{séquentiel}	HGA _{distribué}	ANT	Plan opérationnel
Nb. total contraintes violées	81	77	99	271
Nb. contraintes 1	36	38	47	92
Nb. contraintes 2	42	35	42	117
Nb. contraintes 3	2	3	8	27
Nb. contraintes 4	1	1	2	10
Nb. contraintes 5	0	0	0	25
Temps d'exécution	27 sec	15 sec	-	-

Tableau II.3 – Comparaison des résultats obtenus en résolvant l'instance correspondant au *Réseau3*.

II.2 Modélisation par contraintes

	HGA _{séquentiel}	HGA _{distribué}	ANT	Plan opérationnel
Nb. total contraintes violées	104	104	151	466
Nb. contraintes 1	59	59	74	157
Nb. contraintes 2	42	41	68	202
Nb. contraintes 3	2	2	6	44
Nb. contraintes 4	1	2	3	25
Nb. contraintes 5	0	0	0	38
Temps d'exécution	1 min 47 sec	17 sec	-	-

Tableau II.4 – Comparaison des résultats obtenus en résolvant l'instance correspondant au Réseau4.

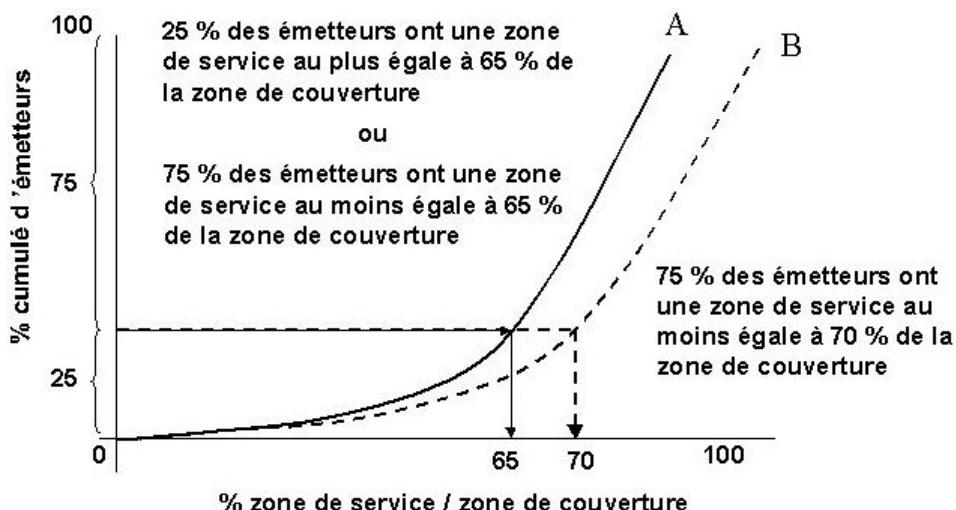


Figure II.2 – Exemple de comparaison de deux solutions.

Afin de mieux comprendre et de lire les résultats fournis par l'outil de validation, nous décrivons le cas l'exemple d'école de la figure II.2. Dans cette figure, nous avons deux courbes :

1. Pour la courbe *A*, deux lectures sont possibles pour un même pourcentage cumulé d'émetteurs. Par exemple, pour 25% des émetteurs :
 - 25% des émetteurs ont une zone de service au plus égale à 65% de la zone de couverture ;
 - 75% des émetteurs ont une zone de service au moins égale à 65% de la zone de couverture.
2. Pour la courbe *B*, on a aussi deux lectures pour 25% des émetteurs. La seconde lecture est : 75% des émetteurs ont une zone de service au moins égale à 70% de la zone de couverture.

La comparaison de la dernière lecture de chaque courbe pour le même pourcentage d'émetteurs (25%) nous permet de constater que la solution correspondant à la courbe *B* est meilleure

Chapitre II. Métaheuristiques pour la planification des réseaux de radiodiffusion

que celle correspondant à la courbe *A*.

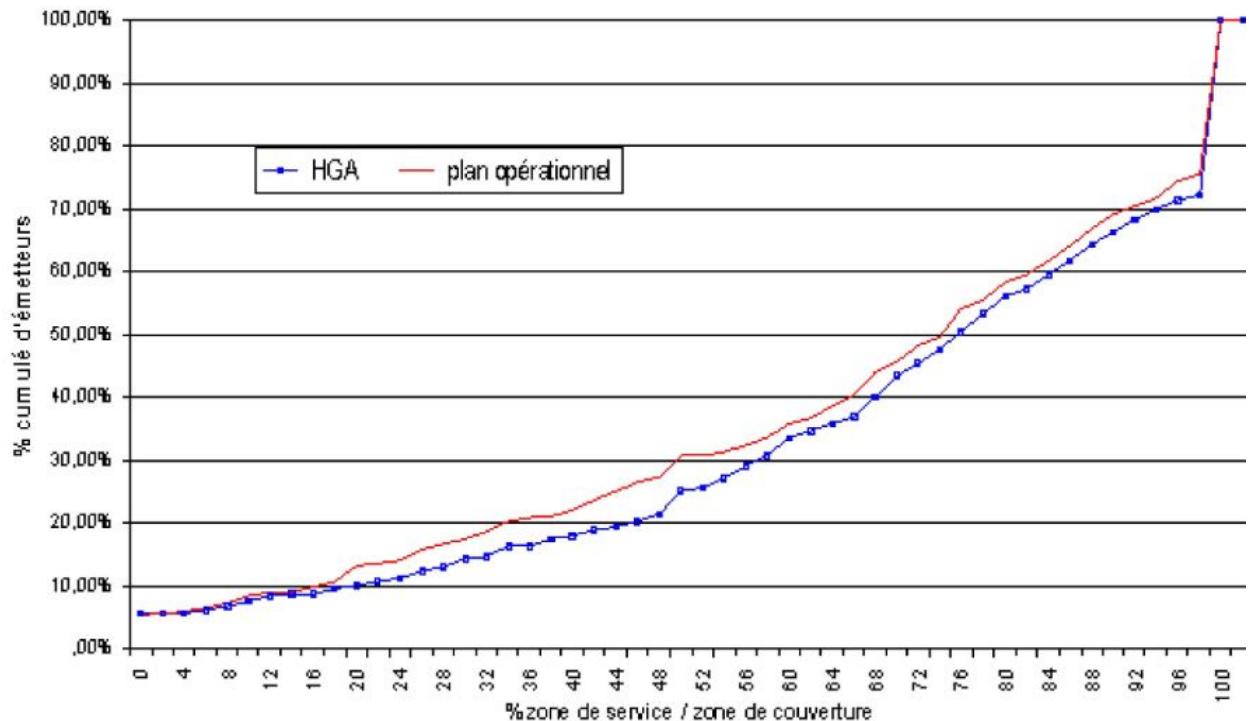


Figure II.3 – Comparaison de la meilleure solution obtenue, pour *Réseau1*, en utilisant *HGA* et la meilleure solution opérationnelle.

La figure II.3 présente la comparaison entre deux solutions en radio *FM* : (1) la solution opérationnelle et (2) une des meilleures solutions obtenues par notre approche (nommée *HGA*).

L'axe des *X* représente le pourcentage $\frac{\text{zone de service}}{\text{zone de couverture}}$. L'axe des *Y* représente le pourcentage d'émetteurs.

Comme nous pouvons le constater sur la courbe de *plan opérationnel* (figure II.3), 80% des émetteurs ont une zone de service au moins égale à 34% de la zone de couverture. Avec notre approche, le même pourcentage d'émetteurs a une zone de service au moins égale à 46% de la zone de couverture. Par conséquent, nous pouvons confirmer que notre solution est de meilleure qualité que la solution opérationnelle. Cette confirmation est d'autant plus justifiée en analysant les courbes comparatives, associées aux plans de fréquence correspondant aux résultats donnés dans les tableaux II.4 et II.3. La figure II.4 montre la comparaison de deux plans de fréquence associés au *Réseau4*. La figure II.5 illustre la comparaison de deux solutions associées au *Réseau3*. Ces différentes figures correspondent à des validations métier faites sur l'ensemble des émetteurs constituant les différents réseaux.

L'analyse de ces différentes courbes de comparaison montre que les plans de fréquence

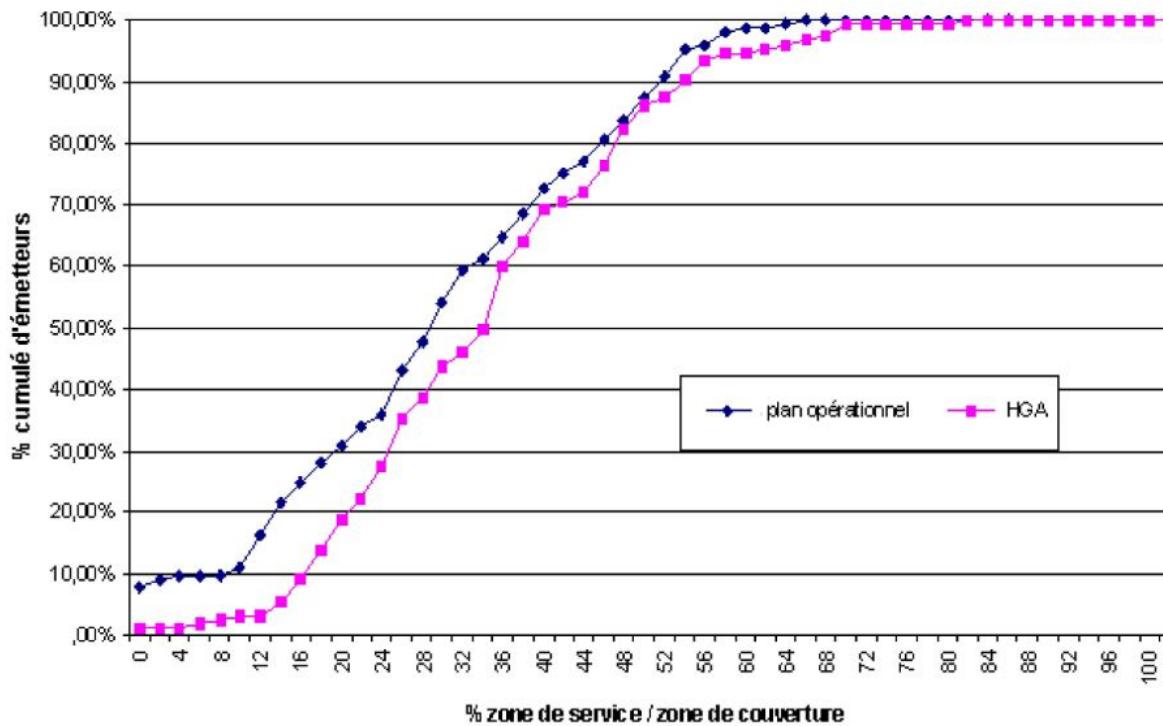


Figure II.4 – Validation de deux solutions trouvées pour le Réseau4.

obtenus par nos approches sont meilleurs en comparaison des plans de fréquence opérationnels en France.

2.4 Discussion

Dans les paragraphes précédents, nous avons décrit l’algorithme génétique hybride que nous avons développé pour résoudre le problème d’allocation de fréquences qui intéresse TDF. Cet algorithme utilise la méthode tabou probabilisée comme opérateur de mutation.

- Les résultats des tableaux et figures présentés permettent de tirer certains enseignements :
- L’analyse des résultats des différentes méthodes données dans ces tableaux nous permet de constater que l’algorithme génétique hybride, nommé *HGA*, se révèle plus performant et satisfait plus de contraintes que les autres approches, telles que la méthode tabou [Coup 99] et un algorithme basé sur une colonie de fourmis [Come 98].
 - Un autre point très intéressant qui peut être déduit de l’analyse des courbes obtenues dans la phase de l’estimation de la qualité des plans de fréquence, est que tous les plans de fréquence obtenus par nos approches sont bons par rapport au plan de fréquence opérationnel dans le cadre de la radiodiffusion en France. En revanche, on peut constater

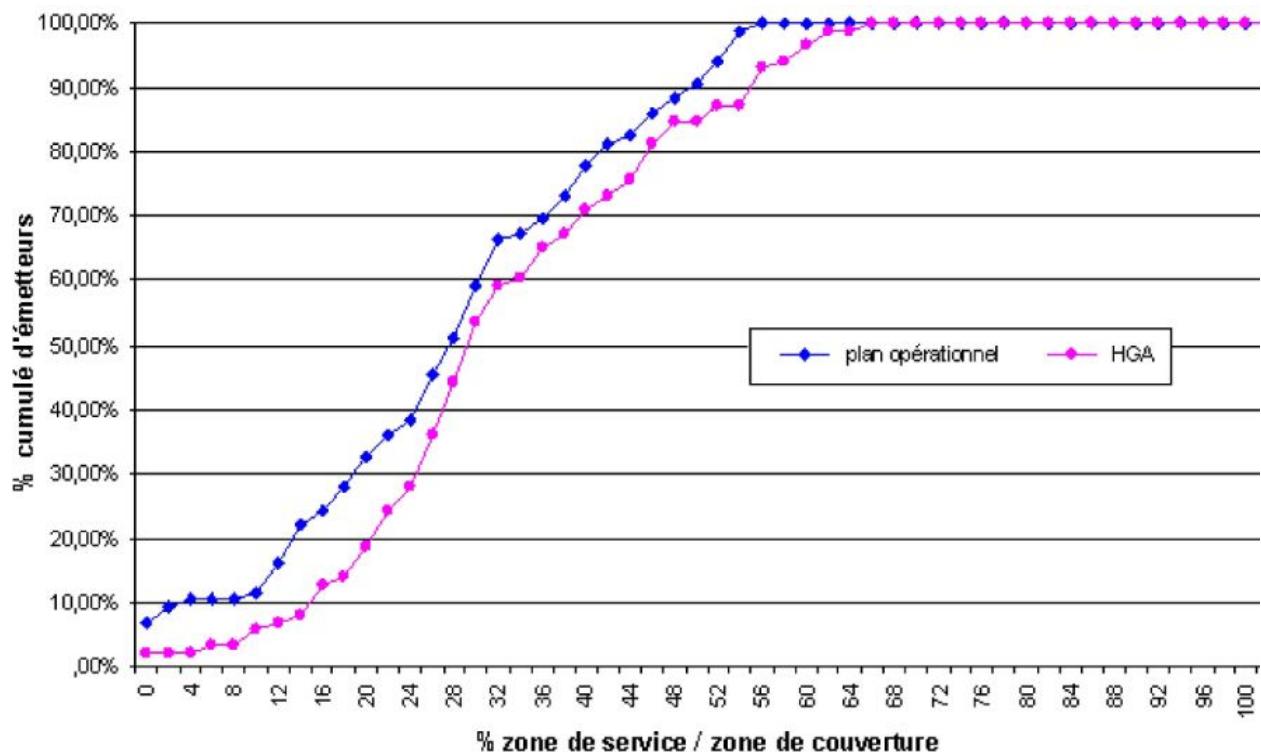


Figure II.5 – Validation de deux solutions trouvées pour le *Réseau3*.

que les écarts entre les mesures de qualité produites par l'outil de validation métier ne sont pas aussi importants que les écarts entre le nombre de contraintes violées par chaque approche.

En effet, ceci est dû aux approximations faites lors de la génération des données d'entrée (matrices de contraintes). Les outils permettant de générer les matrices de contraintes utilisent des modèles de calcul [VHF 95] plus au moins précis². Les approximations utilisées par ces modèles rendent les contraintes plus au moins représentatives. Or, tous nos algorithmes, et de même que d'autres algorithmes utilisés dans ce domaine, utilisent des fonctions objectifs qui supposent que les contraintes sont représentatives, dans le but d'améliorer la qualité des solutions obtenues.

Cette constatation nous a donc conduits à la conclusion suivante : étant donné que l'on dispose d'un algorithme permettant de satisfaire plus de contraintes et dans le but d'améliorer la qualité métier de nos solutions, il est nécessaire de travailler sur l'intégration plus poussée d'informations métier dans le processus d'allocation de fréquences. Par conséquent, il faut développer et mettre en place de nouvelles fonctions objectifs. Les résultats de cette réflexion sont

2. Non disponibilité de modèle numérique de terrain d'une zone frontalière, par exemple.

présentés dans la section 3.

3 Modélisation par histogramme

Dans le but d'améliorer la qualité des plans de fréquence obtenus, il est nécessaire de mettre en place une bonne modélisation des données, qui a pour objectif de représenter plus fidèlement les aspects qui intéressent les planificateurs. Dans cette optique, les réflexions menées, en collaboration avec des équipes de TDF et notamment les réflexions de *Philippe Debreux*³, ont conduit à la mise en place de la modélisation des données, qui est décrite dans cette section.

La modélisation des données d'entrée

L'évolution des outils de planification et l'amélioration de leurs performances passe par une meilleure description des contraintes qui lient les émetteurs entre eux. Ce paragraphe vise à introduire la notion d'histogramme de C/I⁴ et à en montrer la richesse par rapport à l'utilisation "classique" de contraintes. Le concept d'"histogrammes composites" est ensuite abordé pour réduire le temps de calcul et/ou la mémoire nécessaire.

Modélisation par histogramme de C/I

Le calcul des contraintes exploite la cartographie des champs rayonnés calculée pour le réseau à l'étude. Il consiste à déterminer tout d'abord, pour chaque émetteur du réseau, l'ensemble des émetteurs pouvant avoir une influence sur sa zone de couverture, puis à quantifier, pour chacun de ces émetteurs potentiellement brouilleurs une "contrainte" de planification, sous la forme d'un histogramme de C/I.

Sélection des émetteurs potentiellement brouilleurs

La figure II.6 illustre la sélection des brouilleurs effectuée avant détermination de la contrainte de planification.

avec :

– E_u : l'émetteur utile considéré ;

3. Chercheur et Ingénieur d'Etudes

4. Rapport de C/I : le rapport entre la puissance du signal utile reçu en provenance d'un émetteur utile E_u et la puissance interférente en provenance d'un émetteur brouilleur E_{bi}

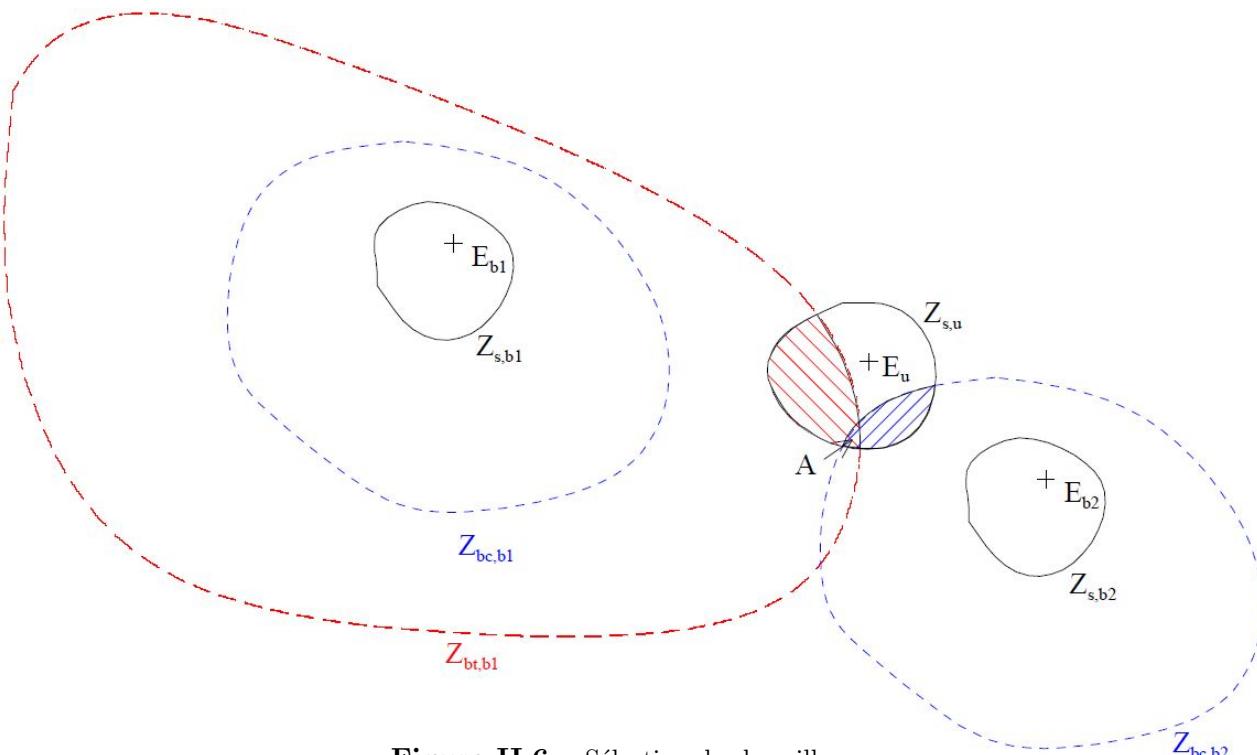


Figure II.6 – Sélection des brouilleurs.

- E_{bi} : l'émetteur brouilleur i ;
- $Z_{s,u}$: la zone de service de l'émetteur utile ;
- $Z_{bc,bi}$: la zone de brouillage constant de l'émetteur brouilleur bi (déterminée par : champ reçu > champ interférent continu admissible) ;
- $Z_{bt,bi}$: la zone de brouillage troposphérique (1% temps⁵) de l'émetteur brouilleur bi (déterminée par : champ reçu > champ interférant troposphérique admissible).

On constate, dans l'exemple ci-dessus, que l'émetteur E_{b1} est un brouilleur troposphérique potentiel de l'émetteur E_u , alors que l'émetteur E_{b2} est un brouilleur continu potentiel. Il est important de noter que les valeurs des champs interférants admissibles (continu ou troposphérique), ainsi que la valeur du champ minimum utilisable, vont évidemment dépendre de la nature des émetteurs utiles et interférants. Notons également que le point marqué A sur la figure II.6 appartient à une zone sur laquelle un phénomène de brouillage multiple devra être

5. **Champ à x% du temps** : un champ radioélectrique suit une distribution *log-normale*. Pour les signaux numériques, c'est la valeur médiane (pour 50% du temps) qui est retenue pour les valeurs de champ utiles et celles à 1% du temps pour les valeurs de champ brouilleurs. Cette valeur de champ brouilleur correspond à un brouillage dit troposphérique. Pour les signaux analogiques, la valeur est la même pour le champ utile ; par contre, pour le champ brouilleur, on retient le champ qui, pondéré par le rapport de protection approprié, est le plus important entre 1% et 50% du temps. Cette distinction permet de ne retenir que le brouillage le plus important entre le brouillage constant et le brouillage troposphérique.

II.3 Modélisation par histogramme

estimé.

Calcul des histogrammes de C/I

Considérant toujours l'exemple de la figure II.6, le calcul des contraintes de planification entre l'émetteur utile E_u et un émetteur potentiellement brouilleur E_{bi} va consister à parcourir la zone de couverture de E_u et à calculer, en chacun de ses points (ou sur une grille d'échantillonnage de ces points), le rapport entre la puissance du signal utile reçu en provenance de E_u et la puissance interférente en provenance de E_{bi} . Ces puissances reçues seront bien sûr calculées en prenant en compte les caractéristiques de l'installation de réception TV (gain et directivité de l'antenne, pertes câble), et en considérant que toutes les antennes de la zone sont orientées vers l'émetteur E_u .

Ce rapport de C/I étant calculé en chaque point de la zone de couverture de l'émetteur utile, il va pouvoir être matérialisé par un histogramme, ou par une fonction de répartition, traduisant l'intensité du brouillage sur cette zone.

La figure II.7 présente un exemple d'analyse de la répartition des brouillages.

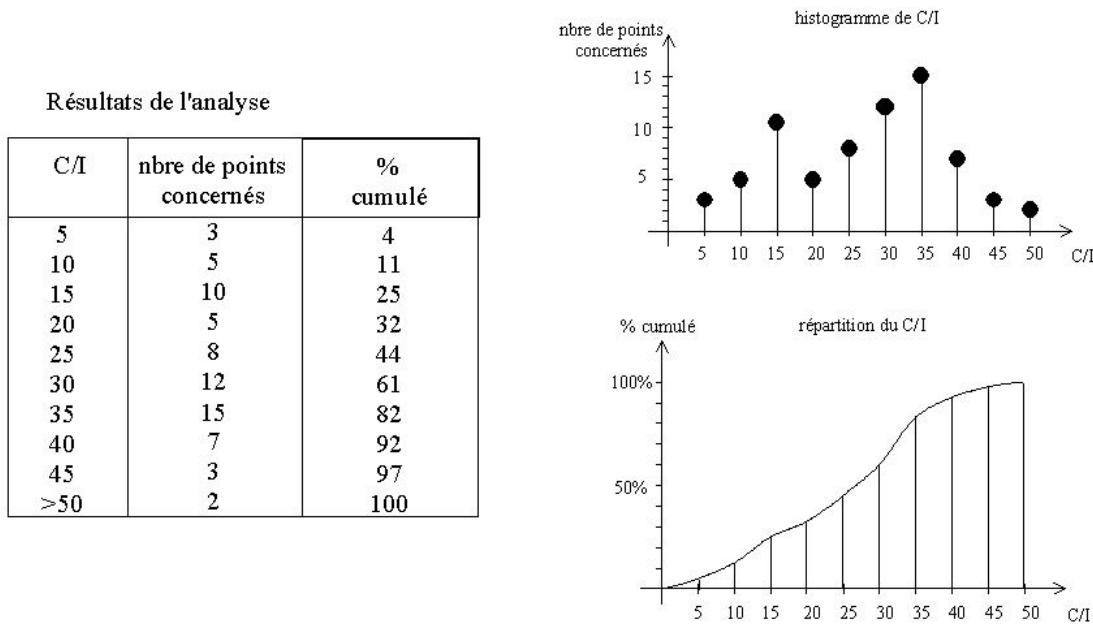


Figure II.7 – Exemple d'histogramme et de fonction de répartition de C/I.

Ce calcul étant fait sur les émetteurs pris deux à deux, les brouillages multiples ne seront pas pris en compte à ce stade, mais dans les étapes suivantes du processus.

À l'issue du calcul, il existe deux histogrammes de C/I : l'un pour les brouillages à 50% du

Chapitre II. Métaheuristiques pour la planification des réseaux de radiodiffusion

temps, l'autre pour les brouillages à 1%. À noter que, pour les émetteurs utiles de TV numérique, seul le second histogramme est calculé.

Exploitation des histogrammes de C/I

L'objectif de la modélisation par histogramme de C/I est de conserver la richesse du calcul de la répartition surfacique des brouillages et ainsi d'évaluer, dès l'allocation, l'impact en terme de nombre de points brouillés du choix d'un écart de fréquences. Ainsi, sur la figure II.8, si l'on considère que le cas traité est celui de deux stations FM, l'affectation d'une fréquence commune, qui nécessite un rapport de protection de 45 dB, induit que seuls 5 points (voir figure II.8) seront couverts sans brouillage. Il s'agit des 5 points qui sont situés à droite de $RP_{cocanal}$. Un écart de 100 kHz entre les deux stations permet de couvrir 27 points, etc.

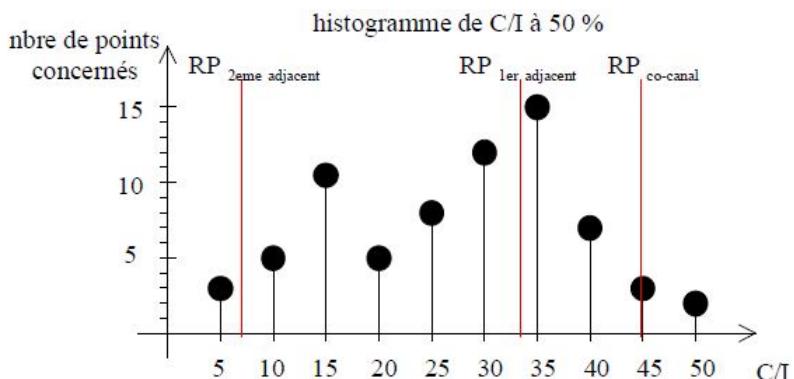


Figure II.8 – Exploitation d'un histogramme de C/I.

Histogramme composite

Si, dans le cas des émetteurs numériques, un seul histogramme est à traiter (brouillage 1% uniquement), en TV analogique ou en FM, le traitement devra s'appuyer sur deux histogrammes. Cette duplicité peut poser des problèmes de gestion de mémoire pour des réseaux de taille importante.

De deux histogrammes de C/I, on peut alors passer à un histogramme où l'axe des abscisses présente autant d'occurrences que de rapports de protection. Pour chaque point traité, les valeurs de $C_I_50\%$ et $C_I_1\%$ sont calculées ; on peut en déduire, grâce aux rapports de protection, la contrainte et l'écart en canaux associés. Il en résulte un histogramme réduit au nombre de rapports de protection, comme illustré dans la figure II.9, et qui peut être interprété de la même façon que précédemment.

II.3 Modélisation par histogramme

Ainsi, pour un point x donné, on a deux valeurs de C/I : $C_I_50\%$ et $C_I_1\%$, on cherche alors à retenir la valeur qui implique le brouillage le plus gênant.

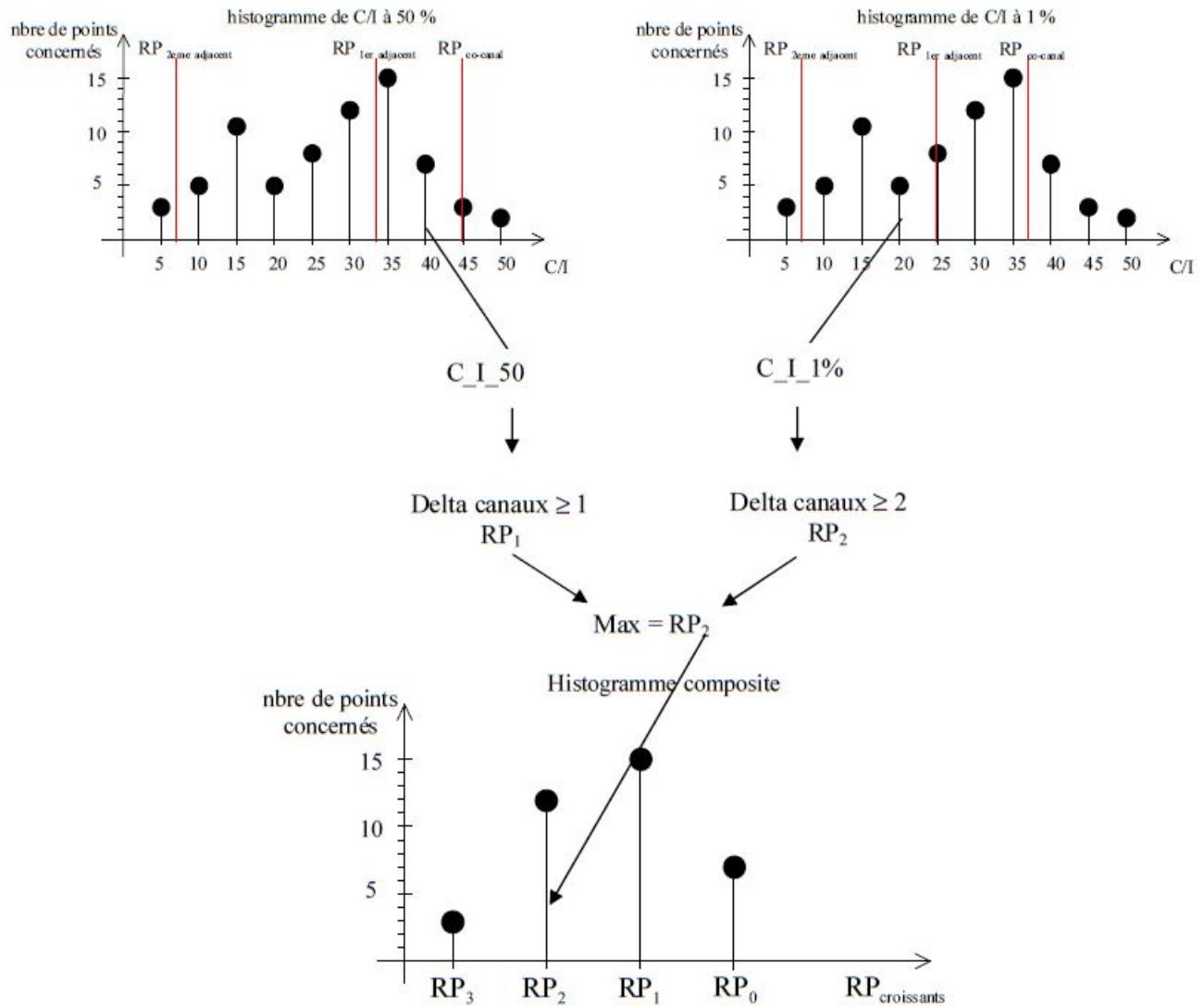


Figure II.9 – Génération de l'histogramme composite.

À noter que, dans la figure II.9 :

- Le classement par rapport de protection croissant ne correspond pas nécessairement à un classement par écart en canaux croissant.
- Le nombre de points de l'histogramme composite correspond à celui des histogrammes de C/I dont il découle ; son exploitation sera similaire ; toutefois, pour l'analyse d'un point de service analogique où deux rapports de protection doivent être respectés (1% et 50%), on s'appuiera sur la valeur de RP la plus forte pour lister les points brouillés.

3.1 Méthode tabou probabilisée

L'algorithme d'allocation de fréquences ([Idou 02e], [Idou 02d]) utilisé est basé sur la méthode tabou probabilisée décrite dans l'algorithme 7. La première étape de cet algorithme consiste à construire le graphe des contraintes. À l'inverse des approches utilisant la modélisation par contraintes, où le graphe est non orienté (voir figure II.10(a)), la modélisation par histogramme impose que le graphe soit orienté, comme le montre l'exemple de la figure II.10(b).

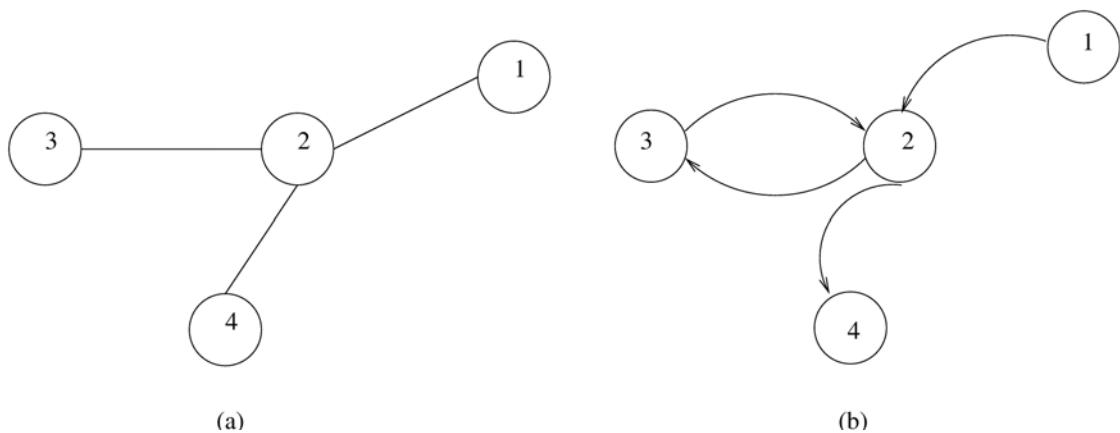


Figure II.10 – Exemples de graphes de contraintes.

Dans ce graphe, l'existence d'un arc ayant comme origine l'émetteur 1 et comme cible l'émetteur 2 signifie que l'émetteur 1 brouille l'émetteur 2. Dans ce cas, on dit que l'émetteur 1 est l'émetteur brouilleur et que l'émetteur 2 est l'utile. À chaque arc est associée une liste établie à partir de l'histogramme de C/I composite. Un exemple de liste manipulée, qui est symétrique dans le cadre de la radio FM, est donné dans la figure II.11.

Décalage canal	4	3	2	1	0	-1	-2	-3	-4
Surface bouillée	0	3	15	123	257	123	15	3	0

Figure II.11 – Structure de la liste utilisée.

Dans cette figure, le *décalage canal* entre un émetteur brouilleur e_b et un émetteur utile e_u est égal à la différence entre la fréquence de e_b et celle de e_u . La *surface brouillée* correspond au nombre de points brouillés pour un écart en canaux donné entre deux émetteurs. Par exemple, supposons que l'on dispose d'un arc A entre un émetteur e_b , qui est l'émetteur brouilleur (source de l'arc), et un émetteur e_u , qui est l'émetteur utile (cible de l'arc). Soit, par exemple, 92 la

II.3 Modélisation par histogramme

fréquence de l'émetteur e_b et 95 celle de l'émetteur e_u . Si la liste décrite dans la figure II.11 est celle qui est associée à l'arc A , alors :

- le décalage canal vaut : $92 - 95 = -3$
- la surface brouillée associée est : 3 points brouillés.

Fonction objectif La fonction objectif a pour but d'évaluer la qualité des solutions et leur adéquation au problème à résoudre. Dans le cadre de l'approche présentée dans ce paragraphe, la fonction objectif utilisée est définie ainsi :

$$F(X) = \sum_{i=1}^n [\sum_{j=1}^n (\phi(e_j, e_i, f_{e_j}, f_{e_i}) \times poids(e_i, e_j) + \psi(e_i, e_j, f_{e_i}, f_{e_j}) \times poids(e_i, e_j))]$$

Avec :

- X : plan de fréquences.
- n : nombre total d'émetteurs constituant le réseau étudié.
- f_{e_j} : fréquence affectée à l'émetteur e_j .
- f_{e_i} : fréquence affectée à l'émetteur e_i .
- $\phi(e_j, e_i, f_{e_j}, f_{e_i})$: nombre de points de la zone de couverture de e_j brouillés par e_i .
- $\psi(e_i, e_j, f_{e_i}, f_{e_j})$: nombre de points de la zone de couverture de e_i brouillés par e_j .
- $poids(e_i, e_j)$: pondération associée aux émetteurs e_i et e_j . Ce coefficient de pondération est fonction du taux de brouillage généré par le choix d'une fréquence (Il est à 1 par défaut dans notre première modélisation).

3.2 Validation métier

Après avoir implémenté l'algorithme tabou probabilisé, nous avons effectué des validations métier sur deux instances réelles de problème d'allocation de fréquences. La première instance correspond au *Réseau5* qui est constitué de plus de 5790 émetteurs et dans lequel on s'intéresse à une replanification de 6 émetteurs. La deuxième instance correspond au *Réseau6* qui est constitué de plus de 5790 émetteurs et dans lequel on s'intéresse à une replanification de 144 émetteurs. Cette limitation est due au temps de calcul très important nécessaire à la génération des listes.

Analyse des résultats

Un phénomène se dégage des premières validations. Supposons que le programme développé ait à choisir entre deux fréquences f_1 et f_2 pour un émetteur replanifiable E , et supposons que le brouillage qui en découle soit celui décrit dans la figure II.12.

Chapitre II. Métaheuristiques pour la planification des réseaux de radiodiffusion

Fréquences possibles pour l'émetteur E	Emetteur impacté par l'affectation d'une fréquence à l'émetteur E				Brouillage total	
	Emetteur X		Emetteur Y			
	Nb points brouillés	Nb points couverts	Nb points brouillés	Nb points couverts	Nb points brouillés	Nb points couverts
f1	200	204	0	2100	200	2304
f2	0	204	400	2100	400	2304

Figure II.12 – Tableau du brouillage entraîné par l'affectation d'une fréquence à l'émetteur E .

Pour faire son choix, le programme compare le nombre de points brouillés par chacune des deux fréquences et retient celle qui brouille le minimum de points. Dans l'exemple ci-dessus, l'affectation de f_1 à l'émetteur E engendre un brouillage total de 200 points sur 2304 points. Par contre, l'affectation de f_2 à l'émetteur E engendre un brouillage total de 400 points sur 2304 points. Étant donné que le but est de minimiser le nombre de points brouillés, c'est donc la fréquence f_1 qui est retenue par le programme. Elle est alors affectée à l'émetteur E . Or, lors de la validation métier, nous avons constaté que ce choix cause un problème. En effet, en choisissant la fréquence f_1 , et non la fréquence f_2 , on se trouve dans la situation suivante :

- pour l'émetteur X on a : 200 points brouillés sur 204 couverts, soit un brouillage de 98% de la zone de couverture de cet émetteur.
- pour l'émetteur Y on a : 0 point brouillé sur 2100 couverts, soit un brouillage de 0% de la zone de couverture de cet émetteur.

Par conséquent, la fréquence affectée à l'émetteur E est mauvaise, car elle dégrade beaucoup les transmissions de l'émetteur X (98% de sa zone de couverture est brouillée). En terme de qualité de validation, il aurait mieux valu choisir d'affecter la fréquence f_2 à l'émetteur E . En effet, cette dernière affectation brouille 0% de la zone de couverture de l'émetteur X et 19% de celle de l'émetteur Y (possédant une zone de couverture beaucoup plus grande).

Cette constatation montre que la fonction objectif favorise les émetteurs puissants (c'est-à-dire ceux qui ont une grande zone de couverture). En effet, elle préfère brouiller complètement la zone de couverture d'un émetteur de faible puissance plutôt que de brouiller partiellement la zone de couverture d'un émetteur plus puissant, car elle minimise ainsi le nombre de points brouillés (dans les cas similaires à celui présenté ici).

Pour remédier à ce phénomène, nous avons introduit un coefficient de pondération qui pénalise le choix d'une fréquence, en attribuant un poids fort lorsque le taux de brouillage d'une zone de couverture est grand.

Comparaison avec des plans de fréquence existants

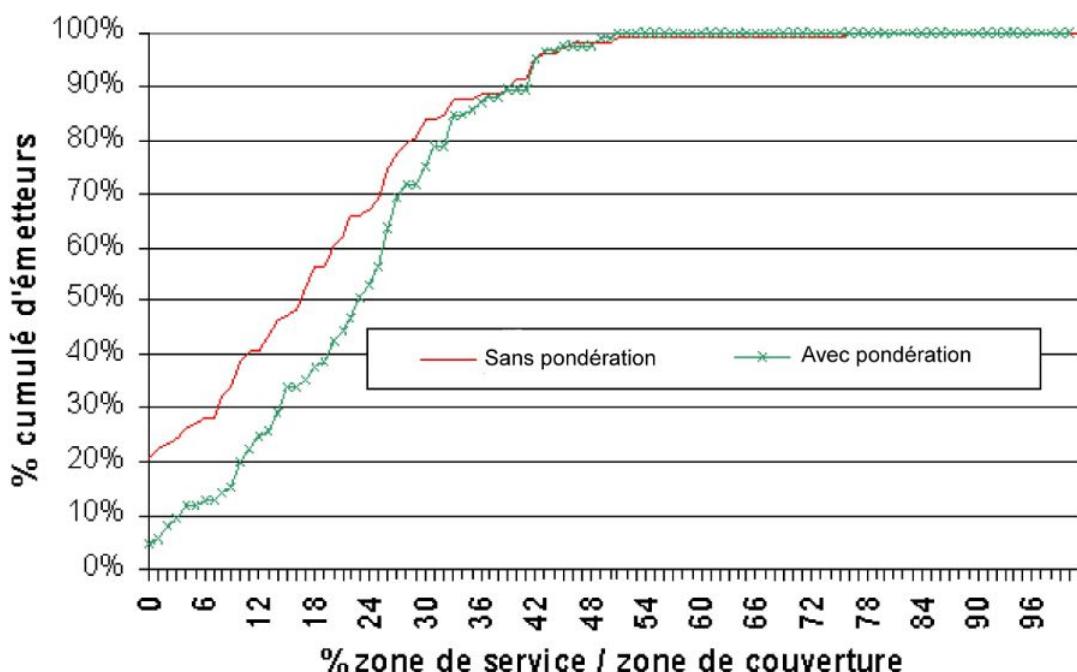


Figure II.13 – Comparaison de deux solutions obtenues en utilisant un coefficient de pondération égal à 1 (*unitaire*) et en pénalisant le choix d'une fréquence en fonction du taux de brouillage engendré.

La comparaison de deux solutions obtenues en utilisant la pondération égale à 1 (*courbe sans pondération*) et en pénalisant le choix d'une fréquence en fonction du taux de brouillage engendré (*courbe avec pondération*) est donnée dans la figure II.13. Nous pouvons constater que l'utilisation de la pondération permet d'améliorer considérablement la qualité des solutions obtenues. Ceci peut être confirmé en analysant les trois courbes de la figure II.14, associées aux trois solutions obtenues en résolvant le problème d'allocation de fréquences lié au Réseau5. Dans cette figure :

- pour la courbe *plan opérationnel*, qui correspond au plan opérationnel en radiodiffusion FM, 70% des émetteurs ont leur zone de service au moins égale à 11% de leur zone de couverture.
- pour la courbe *HGA*, qui correspond à la meilleure solution obtenue en utilisant l'algorithme génétique hybride *HGA*, 70% des émetteurs ont leur zone de service au moins égale à 14% de leur zone de couverture.
- pour la courbe histogramme avec pondération, 70% des émetteurs ont leur zone de service au moins égale à 21% de leur zone de couverture.

Les résultats illustrés dans cette figure montrent que la solution obtenue par l'approche

Chapitre II. Métaheuristiques pour la planification des réseaux de radiodiffusion

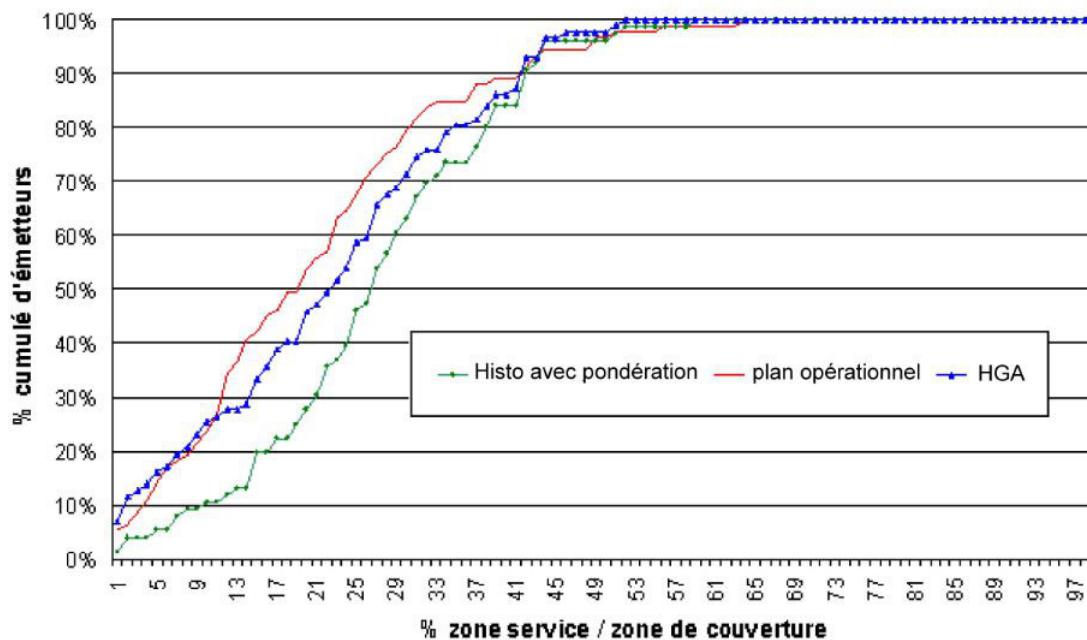


Figure II.14 – Comparaison des différents plans de fréquences pour le Réseau5.

histogramme est de meilleure qualité, comparée au plan de fréquence opérationnel ainsi qu'à la meilleure solution obtenue grâce à l'algorithme génétique hybride *HGA*.

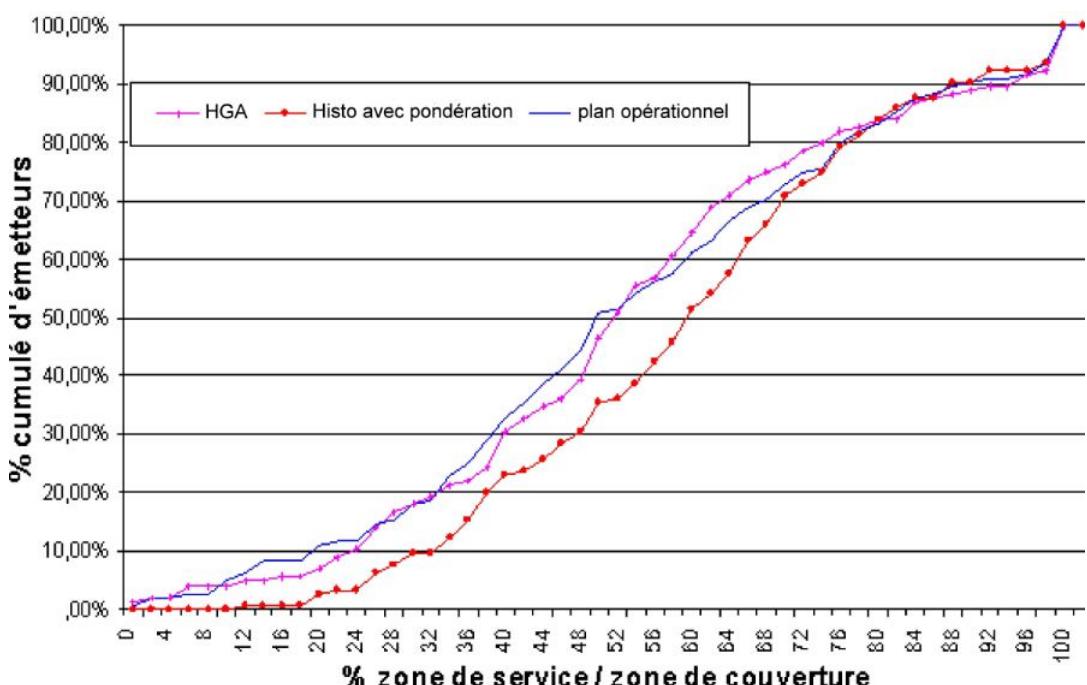


Figure II.15 – Comparaison des résultats obtenus pour le Réseau6.

La figure II.15 présente la comparaison de trois solutions obtenues en résolvant le problème lié au Réseau6. Les courbes correspondent respectivement au plan de fréquence opérationnel (courbe *plan opérationnel*), à la meilleure solution obtenue en utilisant l'algorithme génétique (*courbe HGA*) et à la meilleure solution obtenue en utilisant l'approche par histogramme (*courbe histogramme avec pondération*). Dans cette figure :

- pour la courbe *plan opérationnel*, 90% des émetteurs ont leur zone de service au moins égale à 20% de leur zone de couverture.
- pour la courbe *HGA*, 90% des émetteurs ont leur zone de service au moins égale à 24% de leur zone de couverture.
- pour la courbe histogramme avec pondération, 90% des émetteurs ont leur zone de service au moins égale à 33% de leur zone de couverture.

La comparaison des différentes courbes confirme que la solution obtenue par l'approche par histogramme est nettement supérieure aux autres solutions.

4 Discussion

Les résultats présentés dans le paragraphe précédent nous confirment bien que l'intégration des aspects métier dans la phase d'allocation de fréquences permet d'améliorer la qualité des solutions obtenues.

Dans le but d'améliorer encore la qualité des solutions produites, certains points sont à explorer :

- réaliser une étude plus approfondie sur les coefficients de pondération.
- utiliser la modélisation par histogramme avec l'algorithme génétique hybride décrit dans la première section de ce chapitre, afin de comparer la qualité des plans de fréquences obtenus par les différentes approches.

Chapitre II. Métaheuristiques pour la planification des réseaux de radiodiffusion

Chapitre III

Algorithmes hybrides pour la gestion intelligente de l'énergie dans les smart grids

Les travaux décrits dans ce chapitre ont donné lieu aux publications suivantes : [Roch 12], [Roch 11a], [Roch 11b], [Roch 11c], [Idou 12].

Nos deux articles joints à ce mémoire sont :

- R. Roche, **L. Idoumghar**, B. Blunier and A. Miraoui, “Optimized Fuel Cell Array Energy Management Using Multi-Agent Systems”, 46th Annual Conference of IEEE Industry Applications Society (IAS 2011), October 9-13, 2011, Orlando, FL, USA (CD-ROM).
- R. Roche, **L. Idoumghar**, B. Blunier and A. Miraoui, “Imperialist Competitive Algorithm for Dynamic Optimization of Economic Dispatch in Power Systems”, International Conference on Artificial Evolution (EA-2011), October 24-26, 2011, Angers, (CD-ROM).

Le travail présenté dans ce chapitre résume une partie des travaux que nous avons menés dans le cadre de la préparation de la thèse de M. Robin Roche que j'ai co-encadré à 33% avec le Prof. A. Miraoui et M. B. Blunier¹ (MCF-HDR).

1 Introduction

Le réchauffement climatique et la raréfaction des sources d'énergie fossiles ont conduit les États à prendre des mesures pour favoriser l'émergence de pratiques propres dans le secteur de l'énergie. À titre d'exemple, l'Union Européenne s'est fixée pour objectifs d'augmenter l'efficacité énergétique de 20%, de réduire les émissions de CO₂ de 20% et de faire passer la part des énergies renouvelables à 20% d'ici 2020. Pour atteindre ces objectifs de lourdes modifications dans les processus de production et consommation de l'énergie sont nécessaires. Au cœur de ces modifications, le réseau électrique est amené à évoluer profondément vers ce que l'on nomme

1. Une pensée particulière à Benjamin, brillant chercheur, qui nous a quittés le 22 Février 2012.

Chapitre III. Gestion intelligente de l'énergie dans les smart grids

“smart grid”, ou réseau électrique intelligent [Simo 11] (voir figure III.1).

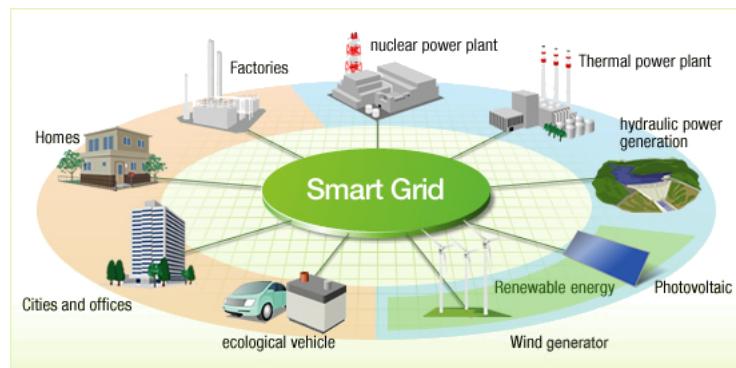


Figure III.1 – Un réseau *smart grid*.

Dans ce réseau modernisé, le réseau électrique est doublé d'un réseau de communication et de contrôle, devant apporter une réelle intelligence à l'ensemble. Il permet d'intégrer les actions de tous les acteurs qui y sont connectés (consommateurs, producteurs, stockeurs), pour fournir de l'énergie électrique de façon durable, économique et sûre.

Les travaux présentés dans ce chapitre s'inscrivent dans cette perspective, et cherchent à lever en partie certains des verrous liés à l'émergence de systèmes de gestion d'énergie intelligents. Ils visent à proposer une solution de gestion d'énergie à la fois efficace et flexible, en alliant les avantages offerts par deux technologies : les systèmes multi-agents, et les algorithmes d'optimisation par métahéuristiques. Des exemples d'applications seront présentés afin d'illustrer les propos développés.

2 Architectures de contrôle décentralisées

La flexibilité du système de gestion proposé est permise par l'utilisation d'un système multi-agents comme base de construction de l'architecture du système.

2.1 Architecture d'un système multi-agents

Un des principaux objectifs du Système de Gestion d'Energie (SGE) est d'atteindre un très haut niveau de flexibilité, non seulement au cours de son fonctionnement, mais aussi pendant les pannes et durant tous son cycle de vie : le système conçu doit être capable de s'adapter à la plupart des changements dans l'architecture du micro-réseau, qu'ils soient intentionnels (par exemple, ajout d'une turbine pour augmenter la capacité de production, ou son arrêt pour la

III.2 Architectures de contrôle décentralisées

maintenance) ou non (par exemple, après une panne). Les systèmes multi-agents, grâce à leurs propriétés et caractéristiques, permettent d'atteindre ce niveau de flexibilité.

2.1.1 Les Systèmes multi-agents

Les systèmes multi-agents (SMA) représentent une forme d'intelligence artificielle distribuée ([Ferb 99], [Hila 08], [Mazi 11]). Ils permettent une approche systémique des problèmes qui peuvent se décomposer en plusieurs entités en interaction. Un SMA est composé de plusieurs entités appelées agents, qui ont leur propre intelligence et autonomie [Ferb 99]. Ces agents situés dans un environnement peuvent interagir en se basant sur leur perception (par exemple, des mesures) et en exécutant des actions (voir figure III.2). Chaque agent a sa propre perception de l'environnement et prend ses décisions sur cette base. En interagissant les uns avec les autres, par exemple grâce à des échanges de messages, les agents peuvent poursuivre leurs propres objectifs et contribuer ainsi à la réalisation du but global visé par le SMA. Ces interactions peuvent être de nature coopératives ou compétitives, c'est-à-dire, ces objectifs peuvent être contradictoires ou non. Les agents peuvent avoir différents niveaux d'intelligence, certains peuvent être dotés d'une intelligence simple tandis que d'autres peuvent disposer de mécanismes de pensée beaucoup plus complexes (par exemple, utilisation des techniques d'apprentissage).

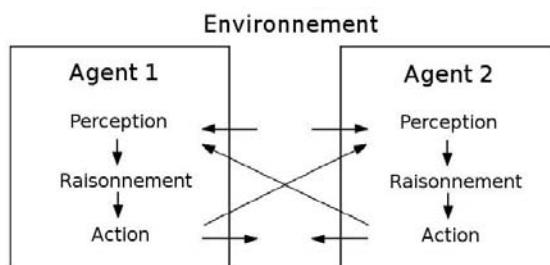


Figure III.2 – Un SMA est constitué d'agents en interaction avec leur environnement.

2.1.2 Intérêt pour un système de gestion d'énergie

Chaque élément composant le réseau électrique peut être considéré comme un agent, avec des rôles, des contraintes, des degrés de perception de l'environnement et moyens d'action propres. Les agents correspondant aux sources, aux moyens de stockage, aux charges électriques, ou encore aux transformateurs, forment donc un SMA qui est l'image du réseau électrique physique.

L'utilisation du concept de SMA dans la problématique des smart grids est motivée par les 3 raisons [Roch 10] suivantes :

Chapitre III. Gestion intelligente de l'énergie dans les smart grids

- De par leur structure distribuée, les SMA sont facilement adaptables à un système distribué dans l'espace tel qu'un réseau électrique. Ceci permet notamment de réduire les besoins en bande passante et en puissance de calcul : les problèmes (coupures, instabilités, etc.) peuvent au moins en partie être traités localement, permettant une plus grande réactivité. Au besoin, ils peuvent également coopérer ou entrer en compétition.
- Les SMA sont flexibles, grâce à leur architecture distribuée, et peuvent donc s'adapter aux évolutions du réseau, qu'il s'agisse de pannes, d'ajout ou de suppression de composants, et ce, que ces évènements aient été prévus à la conception du système ou non. Ils sont donc tolérants aux fautes et permettent un fonctionnement dégradé et *plug & play*.
- Les SMA sont proactifs, en cherchant à satisfaire leurs besoins ou objectifs en fonction de leurs rôles et contraintes. Une charge cherchera donc à être alimentée, tout comme un gestionnaire de centrale cherchera à minimiser ses coûts tout en respectant ses contraintes opérationnelles et légales. En cas de besoin, les agents peuvent également aller chercher les informations dont ils ont besoin pour prendre des décisions et planifier des actions.

On remarque donc une adéquation entre les caractéristiques des SMA et celles requises par les smart grids, montrant l'intérêt de leur utilisation.

2.1.3 Architecture sélectionnée

Dans un SMA, les agents peuvent être organisés de nombreuses manières : équipes, hiérarchies, coalitions, fédérations, marchés, etc. Dans ce travail nous avons opté pour une forme de structure en fédérations. Celle-ci est la plus proche de la structure que l'on peut imaginer que les smart grids prendront à terme, à savoir un ensemble de micro-réseaux partiellement autonomes, interconnectés entre-eux et s'échangeant de l'énergie. La Figure III.3 l'illustre, et présente plusieurs micro-réseaux. À cet ensemble de micro-réseaux on rajoute un système de gestion, hiérarchisé, qui sera à la fois une interface de coordination des composants et un acteur de *trading* d'énergie avec les autres systèmes de gestion.

L'architecture de SMA proposée (figure III.4) se compose de plusieurs agents, répartis en trois catégories :

- Agents de contrôle des composants physiques : turbines, stockage, charge et réseau. Un agent *turbine* est créé pour chaque turbine dans le micro-réseau. L'agent réside dans le système de contrôle propre à la turbine. Il communique avec l'agent SCADA²en lui envoyant son état actuel grâce à ses mesures, et reçoit des consignes en retour. Cet agent est le seul à avoir accès à toutes les données disponibles sur la turbine. De même pour les

2. Supervisory Control And Data Acquisition

III.2 Architectures de contrôle décentralisées

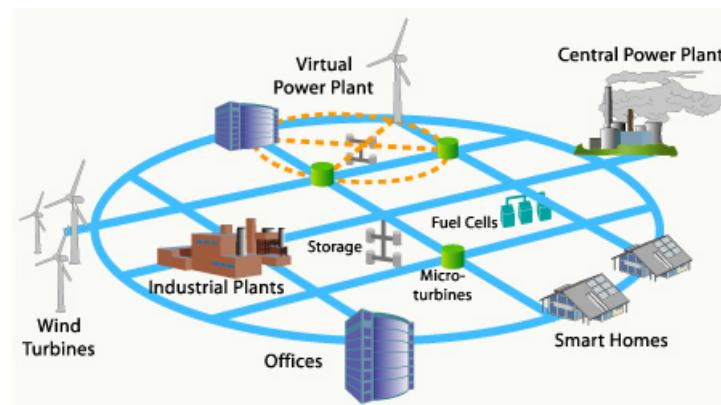


Figure III.3 – Exemple de plusieurs micro-réseaux qui forment un smart grid.

autres agents, ils peuvent dialoguer avec l’agent *SCADA* en lui envoyant leurs mesures actuelles et reçoivent des recommandations en retour.

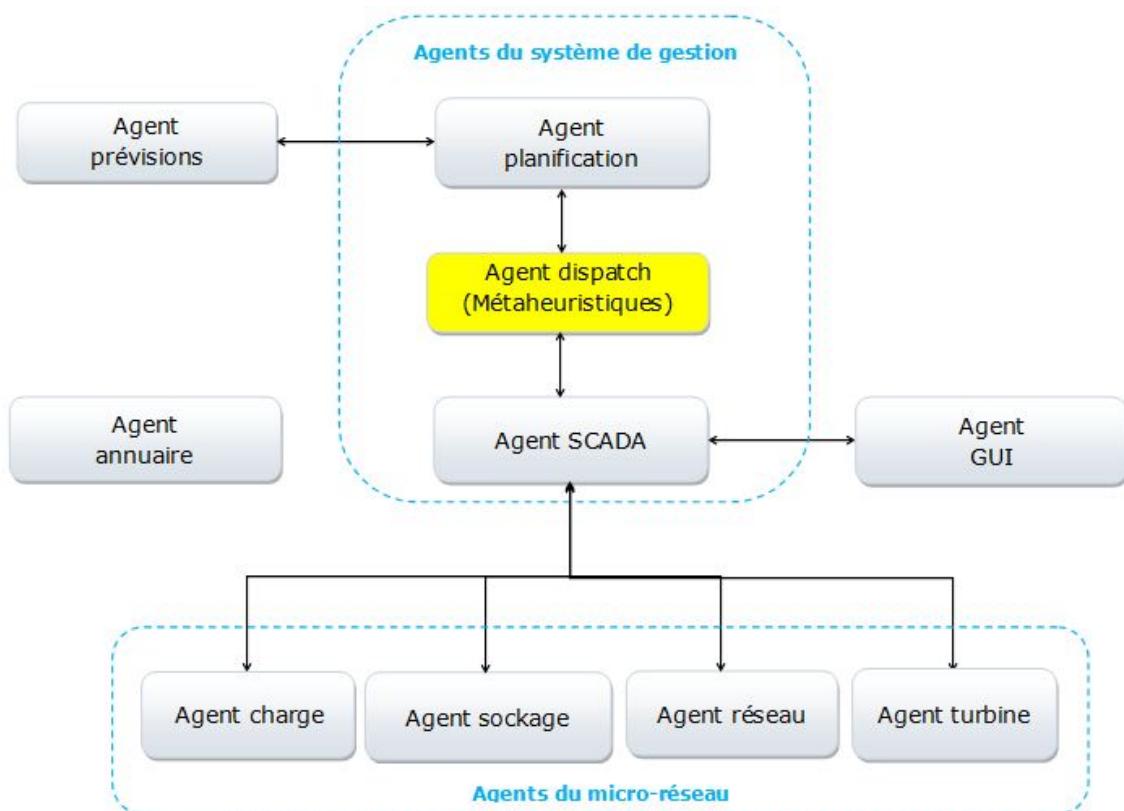


Figure III.4 – Le système multi-agents proposé est constitué de plusieurs types d’agents, avec un fonctionnement propre à chacun d’entre eux. Chaque flèche correspond à un canal de communication habituel entre les agents en mode de fonctionnement normal.

- Agents du système de gestion : *SCADA*, *dispatching* et *planification*. L'agent *SCADA* (système d'acquisition et de supervision) joue le rôle d'un superviseur et contrôleur commercial. Il recueille les données des différents agents qui forment le micro-réseau, et leur transmet les consignes reçus auprès de l'agent chargé de dispatching. Il fournit également aux autres agents les informations dont ils ont besoin pour fonctionner, à chaque fois qu'ils le demandent et si ils sont autorisés à le faire. L'agent *dispatch* calcule, à partir des données reçues de l'agent *SCADA*, le point de fonctionnement optimal pour chaque agent du micro-réseau. Il maintient également une base de données contenant les informations dont il a besoin pour déterminer la bonne répartition des puissances en utilisant des données économiques et environnementales. L'agent *planification* est chargé de la planification à l'avance, si possible optimale, du fonctionnement de l'ensemble, à l'aide de prévisions de production, de demande et de prix émis par des agents.
- Agents auxiliaires : formés principalement de l'agent *annuaire* et *GUI*. L'agent *annuaire* sert de serveur de noms et de services, il fournit aux agents qui en font la demande les noms et adresses des agents correspondant à leurs besoins (tous les agents peuvent communiquer avec lui). L'agent *GUI*, gère l'interface homme-machine, reçoit les mises à jour sur l'état actuel du système, et permet aux opérateurs humains d'assurer la supervision. Il transmet également les commandes émises par les opérateurs à l'agent *SCADA*.

2.1.4 Structure d'un agent

Tous les agents sont basés sur une structure générique (Figure III.5). Quand un agent est créé, il exécute un ensemble d'instructions d'initialisation. Ces instructions lui permettent d'obtenir des informations sur le composant dont il est en charge (par exemple la turbine), et de s'inscrire auprès des autres agents avec lesquels il interagira. Il se met en mode d'attente d'un message en provenance d'un autre agent. Ce message peut être une demande ou peut contenir une information dont l'agent aurait besoin. En fonction du contenu du message, l'agent exécute une séquence donnée d'instructions, comme par exemple lancer des mesures, interagir avec les autres agents. A l'issu de cette séquence, l'agent se remet en mode d'attente d'un message, et peut exécuter des tâches d'arrière-plan jusqu'à ce qu'un nouveau message soit reçu. Si le message indique à l'agent de s'arrêter, l'agent exécute quelques instructions de nettoyage et s'arrête après s'être désinscrit auprès des autres agents. Lorsqu'un problème de communication est détecté, l'agent a également la possibilité de basculer en mode dégradé, dans lequel il peut choisir de s'arrêter progressivement pour assurer la sécurité du système, s'il ne parvient pas à rétablir la communication.

III.2 Architectures de contrôle décentralisées

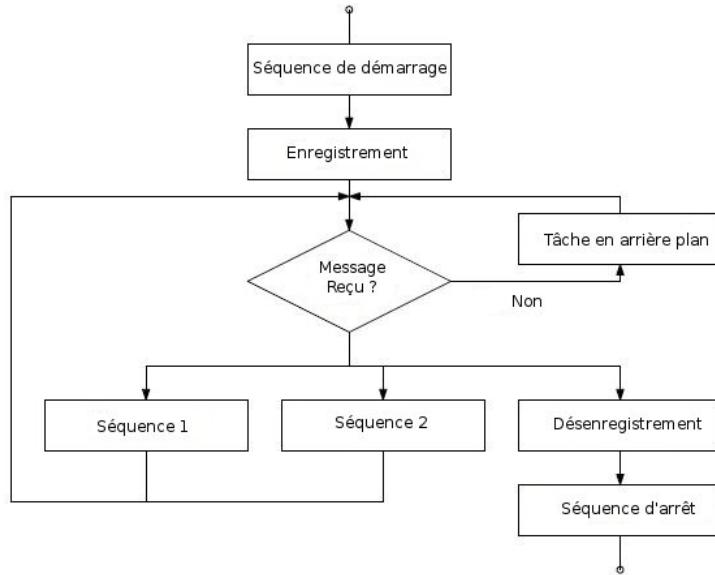


Figure III.5 – Diagramme générique de cycle de vie d'un agent, de sa création à son arrêt. Les séquences d'instructions sont choisies en fonction des interactions avec les autres agents.

2.2 Interactions entre les agents

Les interactions des agents dans notre SMA dépendent de l'évolution du système. En effet, deux situations se présentent : la modification de l'architecture et le fonctionnement normal du système.

2.2.1 Changement dans l'architecture SMA

Lors du changement de l'architecture du système, par exemple, au démarrage ou lorsqu'une turbine est ajoutée, la structure de SMA a besoin de s'adapter. Dans le premier cas, l'agent *GUI* et l'agent du système de gestion sont lancés, initialisés et paramétrés par l'opérateur. Le système attend alors que les agents micro-réseau s'inscrivent. Le fonctionnement du système peut alors commencer. Pour un agent type turbine, il suit le processus décrit ci-après (voir les trois premiers messages de la figure III.6) :

1. L'agent de la turbine est créé, comme *turbine*, et son système de contrôle est démarré.
2. Pour son initialisation, il charge ou mesure les caractéristiques de la turbine (modèle, puissance max en sortie, type de carburant, etc.).
3. Il s'inscrit auprès de l'agent *annuaire*, pour permettre aux autres agents de le trouver dans le cas où ils auront besoin d'une turbine.

4. Il s'enregistre ensuite auprès de l'agent *SCADA*, et attend une demande pour transmettre des informations sur son état actuel. L'agent a alors atteint son état normal de fonctionnement.

5. L'agent *SCADA* enregistre enfin le nouvel agent auprès de l'agent *dispatch*. La turbine est désormais pleinement opérationnelle.

De même, quand il faut déconnecter un composant, soit par l'opérateur ou par le système, l'agent correspondant commence par se désinscrire auprès de l'agent *annuaire* et des bases de données des autres agents, de sorte qu'ils ne tentent plus de communiquer avec lui. Une série d'instructions relatives à l'arrêt de l'agent, par exemple l'enregistrement des données mesurées, sont ensuite exécutées avant son retrait effectif du système. Dans le cas où l'agent doit être temporairement déconnecté, l'agent peut être suspendu en passant à un mode *veille*. En cas de besoin, l'agent peut "rebasculer" en mode *normal*. Cette capacité du système à s'adapter aux changements dans sa structure lui permet de s'adapter à une variété très large de micro-réseaux, avec un nombre variable de composants (turbine, piles à combustion, etc.).

2.2.2 Fonctionnement normal du système

Dans un fonctionnement normal du système, c'est-à-dire lorsque la structure du système ne change pas, le système exécute les mêmes instructions en permanence à une fréquence donnée choisie par l'opérateur (voir les étapes 6 à 14 de la figure III.6) :

6. L'agent *dispatch* demande à l'agent *SCADA* de lui transmettre les mesures sur l'état actuel de l'ensemble du système.
7. L'agent *SCADA* demande à chaque agent, inscrit dans sa base de données, de lui transmettre les dernières informations de son état, par exemple les mesures en cours.
8. Chaque agent, réalise ses mesures et mis à jour ses caractéristiques.
9. Les données mises à jour sont ensuite envoyées à l'agent *SCADA*, qui centralise les mesures et met à jour sa propre base de données.
10. L'agent *SCADA* envoie les données demandées à l'agent *dispatch*.
11. L'agent *dispatch* calcule les points de fonctionnement optimaux.
12. Ces points de fonctionnement sont enfin envoyés à l'agent *SCADA*, qui se charge de les transmettre à leurs destinataires respectifs.
13. Les résultats sont également envoyés à l'agent *GUI*, qui les affiche.

III.3 Optimisation du fonctionnement des micro-réseaux et des centrales

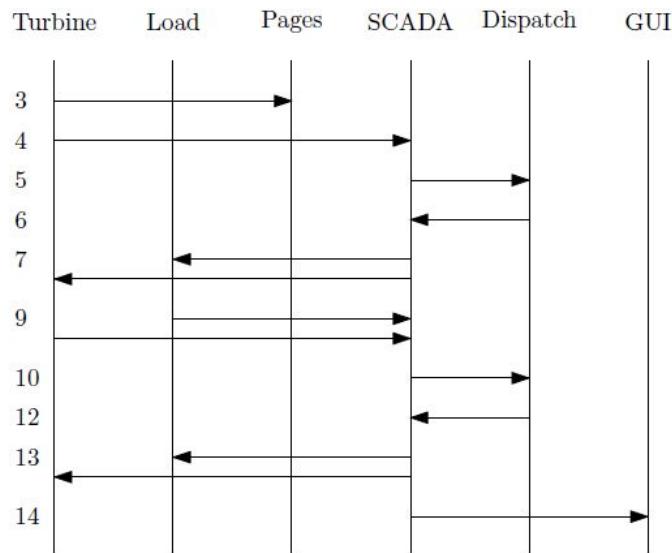


Figure III.6 – Durant leurs opérations, les agents interagissent les uns avec les autres en échangeant des informations par le biais des messages. Chaque flèche décrit la direction de transmission des messages entre deux agents. Cet exemple est limité à deux composantes, une turbine et la charge, pour faciliter la compréhension. Les étapes 3 à 5 se réfèrent à la section 2.2.1 et les étapes 6 à 13 se réfèrent à la section 2.2.2.

3 Optimisation du fonctionnement des micro-réseaux et des centrales

L'efficacité du système est quant à elle obtenue grâce à des algorithmes d'optimisation basés sur des métaheuristiques. Ils sont au cœur des deux couches supérieures du système de gestion de la figure III.4 et leur donnent leur intelligence.

3.1 Algorithmes d'optimisation

3.1.1 Algorithme *MPSOM* (*Metropolis Particle Swarm Optimization with Mutation operator*)

Dans cette sous-section, nous présentons l'algorithme hybride que nous avons proposé dans [Idou 10b] et qui est noté *MPSOM*. Cet algorithme se base sur l'algorithme *PSO* auquel on incorpore un opérateur de mutation et la règle de *Metropolis* afin de sortir d'un optimum local. Le principe de *MPSOM* fonctionne comme illustré dans l'algorithme 8, où :

- **Particule** : chaque particule (solution) est représentée par sa :

Algorithme 8: Algorithme hybride *MPSOM*

```

1 Initialiser swarm_size particules
2 Evaluer (essaim)
3 critère d'arrêt  $\leftarrow$  nombre maximum d'évaluations de la fonction objectif
4 AucuneAmélioration  $\leftarrow 0$ 
5 Initialiser le facteur d'inertie  $w \leftarrow w_0$ 
6 Initialiser la température initiale  $T \leftarrow T_0$ 
7 Tant que critère d'arrêt n'est pas vérifié Faire
8   Trier l'essaim
9   Si AucuneAmélioration < k Alors
10    | Pour toute particule  $i \leftarrow 1$  à swarm_size Faire
11    |   Acceptation ( $X$ ,  $c_{best}$ ,  $T$ )
12    |   Mise à jour de la vitesse en utilisant l'équation (III.2)
13    |   Confinement de la vitesse
14    |   Mise à jour de la position de la particule en utilisant l'équation (III.5)
15    |   Confinement de la position
16    | Fin
17   Sinon
18    | //Opérateur de mutation
19    | AucuneAmélioration  $\leftarrow 0$ 
20    | Pour toute particule  $i \leftarrow 1$  à swarm_size Faire
21    |   | Initialiser sa vitesse à la vitesse maximale autorisée
22    | Fin
23 Fin
24 Evaluer (essaim)
25 Si aucune amélioration de la solution globale Alors
26   | AucuneAmélioration  $\leftarrow$  AucuneAmélioration + 1
27 Sinon
28   | Mettre à jour de la solution globale
29   | AucuneAmélioration  $\leftarrow 0$ 
30 Fin
31 Mettre à jour la température  $T$ 
32 Mettre à jour le critère d'arrêt
33 Mettre à jour du facteur d'inertie  $w$  en utilisant l'équation (III.4)
34 Fin

```

- position courante $X \in S$ représentée par ses $n > 0$ composants, i.e. $X = (x_1, x_2, \dots, x_n)$ où n représente la dimension du problème d'optimisation à résoudre.
- meilleure solution obtenue précédemment (appelée c_{best}).
- vitesse v qui correspond au taux de changement de la position.

III.3 Optimisation du fonctionnement des micro-réseaux et des centrales

- **Essaim initial** : correspond à la population de particules qui vont évoluer. Chaque particule x_i est initialisée avec une valeur uniformément tirée entre les bornes inférieure et supérieure de l'intervalle définissant le problème d'optimisation.
- **Fonction Evaluer** : correspond à la fonction objectif que l'algorithme *MPSOM* doit minimiser.
- **Tri** : toutes les particules de l'essaim sont triées dans l'ordre décroissant de leur fonction objectif.
- **Fonction Acceptation** : $Acceptation(cbest, X, T)$ est déterminée par la probabilité d'acceptation donnée par l'équation III.1, qui est la probabilité d'accepter la position courante d'une particule comme étant son *cbest*.

$$p = \begin{cases} 1 & \text{si } f(X) \leq f(cbest) \text{ ou} \\ & \frac{1}{2}rand \times (1 + e^{\frac{f(X)-f(cbest)}{T}}) < 1 \\ 0 & \text{sinon} \end{cases} \quad (\text{III.1})$$

Dans l'équation III.1, T est la température. Elle joue le même rôle que la température dans la méthode du recuit simulé (section 2.3.1). $rand$ est un nombre aléatoire indépendamment généré dans l'intervalle $[0, 1]$.

- **Mise à jour de la vitesse** : nous proposons ici, un voisinage topologique qui est déterminé dynamiquement durant le processus de recherche selon la fonction objectif de l'essaim. Après avoir trié la population de l'essaim pour chaque $i^{\text{ème}}$ particule, un ensemble N_i de ses voisins est défini comme : $N_i = \{\text{particule } k / k \geq i \text{ et } k \leq \text{swarm_size}\}$. Ensuite, afin d'ajuster la vitesse de la $i^{\text{ème}}$ particule, nous utilisons l'équation III.2 suivante :

$$\begin{aligned} v_{ij} = & \omega \times v_{ij} + c_1 \times rand \times (cbest_{ij} - x_{ij}) \\ & + min(V_{max}, \sum_{k=i}^{\text{swarm_size}} \frac{cbest_{kj} - x_{ij}}{k}) \end{aligned} \quad (\text{III.2})$$

Dans l'équation III.2, le composant social de la $i^{\text{ème}}$ particule est calculé comme étant la moyenne pondérée des meilleures particules dans N_i . En procédant de la sorte, une particule n'est pas seulement influencée par la meilleure solution globale, mais elle ajuste sa vitesse en fonction de la moyenne pondérée des solutions qui sont meilleures que les siennes. Cela signifie simplement qu'il est mieux pour la particule de suivre un groupe de particules plutôt que de suivre une seule particule.

- **Mise à jour de la température** : afin d'éviter à l'algorithme d'être coincé dans un minimum local, le taux de réduction de la température doit être lent. Nous utilisons pour

cela l'équation III.3 où $i = 0, 1, \dots$ et $\gamma = 0.99$.

$$T_{i+1} = \gamma T_i \quad (\text{III.3})$$

- **Mise à jour du facteur d'inertie** : le facteur d'inertie est défini par l'équation III.4 où :

$$w_k = w_0 \times \left(1 - \frac{T_0 - T_k}{T_0}\right) \quad (\text{III.4})$$

- $w_0 = 0.9$ correspond à la valeur du poids au démarrage de l'algorithme. Notons qu'avec de faibles valeurs de w , la région de recherche de l'algorithme peut se trouver autour de la meilleure solution, alors qu'avec des valeurs élevées de w , l'algorithme peut améliorer l'exploration (recherche globale).
- T_0 est une température initiale et T_k représente la température à la $k^{i\text{eme}}$ itération.
- **Opérateur de mutation** : s'il n'y a pas d'améliorations de la meilleure solution globale pendant les dernières K itérations, cela signifie que l'algorithme est coincé dans un optimum local. Afin d'en échapper, notre algorithme *MPSOM* utilise l'opérateur de mutation en se basant sur l'idée suivante : en attribuant une vitesse maximale permise à chaque particule, l'équation III.5 assure que toutes les particules vont sortir de l'optimum local et *MPSOM* peut avoir une large capacité d'exploration.

$$x_{i+1} = x_i + v_{i+1} \quad (\text{III.5})$$

3.1.2 Principe de l'algorithme ICA (*Imperialist Competitive Algorithm*)

ICA (*Imperialist Competitive Algorithm*) est une approche récente développée par E. Atashpaz Gargari et Caro Lucas [Atas 07]. Elle tente de reproduire le comportement des empires. Il met en place un mécanisme de compétition afin d'affaiblir puis de supprimer les empires les moins puissants (c'est-à-dire les empires trouvant les moins bonnes solutions).

Dans cette section nous allons décrire le fonctionnement de l'algorithme ICA. Imperialist Competitive Algorithm (ICA) est une méthode qui utilise l'impérialisme et la compétition impérialiste comme source d'inspiration. L'algorithme 9 décrit le fonctionnement de l'algorithme ICA dont les principales étapes sont :

- **Initialisation** : dans cet algorithme chaque individu de la population représente un pays, la première étape est de les initialiser puis de les évaluer. Les meilleurs pays sont sélectionnés et deviennent les impérialistes et ceux qui ne le sont pas deviennent les

III.3 Optimisation du fonctionnement des micro-réseaux et des centrales

Algorithme 9: Algorithme ICA (Imperialist Competitive Algorithm)

```
1 Initialiser et évaluer les empires
2 Tant que critère d'arrêt n'est pas vérifié faire
3   Mouvement des colonies vers leurs impérialistes
4   Révolution. Faire quelques changements dans les caractéristiques de certains des pays
5   Si il y a une colonie dans un empire qui a meilleur coût que celui de son impérialiste
      Alors
        | Echanger les positions de la colonie et de l'impérialiste
      Fin
5   Calculer le coût total de tous les empires
6   Si la distance entre deux empires est inférieure au seuil de fusion Alors
7     | fusionner les deux empires
8   Fin
9   Compétition impérialiste
10  Si un empire est sans colonies Alors
11    | Eliminer cet empire
12  Fin
13 Fin
```

colonies des impérialistes. Les colonies sont ensuite réparties sur les empires en fonction de leur puissance.

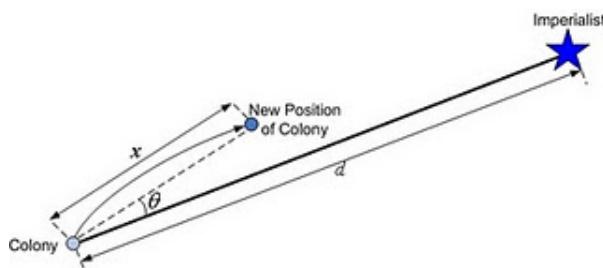


Figure III.7 – Déplacement d'une colonie

- **Mouvement :** après avoir réparti les colonies entre les différents impérialistes, chaque colonie se déplace en direction de son impérialiste. La figure III.7³ illustre ce mouvement, d est la distance entre la colonie et l'impérialiste. θ et x sont des nombres aléatoires. Le nombre θ , tiré aléatoirement, permet de faire dévier légèrement la colonie. Après chaque mouvement d'une colonie, il se peut que le coût d'une colonie soit inférieur au coût de son impérialiste, dans ce cas, la colonie devient l'impérialiste.
- **Coût total d'un empire :** le coût total d'un empire dépend du coût de l'impérialiste et

3. Figure extraite de <http://www.atashpaz.com/p/imperialist-competitive-algorithm-ica.html>

de ses colonies. La formule (III.6) donne l'expression qui permet de calculer le coût total d'un empire n .

$$TC_n = Cost(Impérialiste_n) + \xi \cdot mean(colonies of empire_n) \quad (\text{III.6})$$

avec ξ est un nombre inférieur à 1.

- **Compétition impérialiste :** la compétition impérialiste permet à un empire plus fort de s'emparer d'une colonie de l'empire le plus faible. Cette compétition permet ainsi d'augmenter le pouvoir des empires forts et d'éliminer les empires faibles.

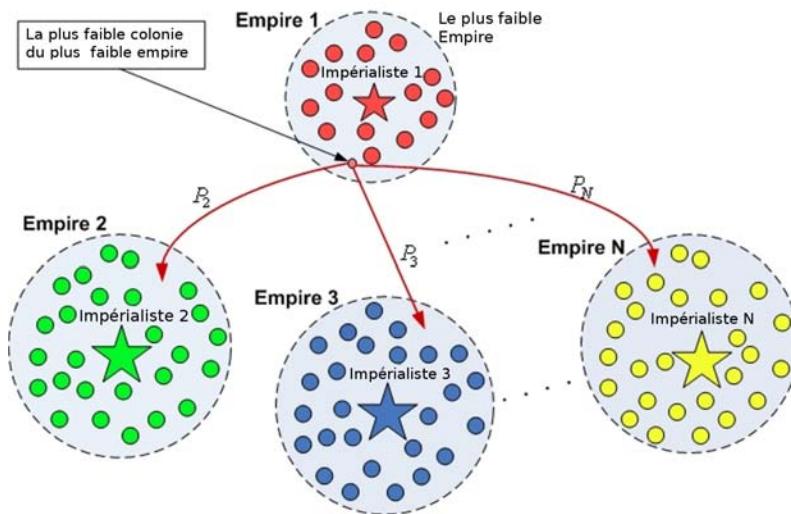


Figure III.8 – Principe de la compétition des empires : l'empire le plus puissant a plus de chance de posséder la plus faible colonie de l'empire le plus faible.

La figure III.8 illustre cette compétition, l'empire 1 le plus faible est sur le point de perdre une colonie au profit d'un des empires. Chaque empire a une probabilité de possession P_i calculée à partir de sa puissance.

$$P_{p_n} = \left| \frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i} \right| \quad (\text{III.7})$$

avec $NTC_n = TC_n - \max_i TC_i$, TC_n et NTC_n correspondent respectivement au coût total et au coût total normalisé du $n^{\text{ème}}$ empire.

Le mouvement des colonies vers l'impérialiste et la compétition permet de voir émerger un état où il n'y a plus qu'un seul empire et où l'algorithme a trouvé idéalement une solution proche de la solution optimale.

III.3 Optimisation du fonctionnement des micro-réseaux et des centrales

3.2 Problème du *dispatching*

La résolution du problème du *dispatching* optimal est réalisée à l'aide des deux algorithmes *MPSOM* [Roch 11b] et [Roch 11a] précédemment détaillés. Le réseau évoluant en permanence, il s'agit donc d'un problème dynamique, contraint et potentiellement à objectifs multiples. Une solution correspondant à un vecteur composé des puissances des composants que l'on souhaite calculer et prendre en compte. L'espace d'optimisation pour chaque dimension est défini par l'agent lui-même, en fonction de ses propres contraintes.

L'objectif principal est généralement le coût de fonctionnement, que l'on cherche à minimiser. Ce coût correspond notamment au coût du combustible et aux achats et ventes d'énergie sur le réseau principal. On cherche donc à minimiser :

$$f(t) = \sum_{i=1}^{n_{gen}} c(P_{gen,i}(t)) + \sum_{i=1}^{n_{enr}} c(P_{enr,i}(t)) + \sum_{i=1}^{n_{sto}} c(P_{gen,i}(t)) + \sum_{i=1}^{n_{res}} c(P_{res,i}(t)) \quad (\text{III.8})$$

Soumis à :

$$\sum_{i=1}^{n_{gen}} P_{gen,i}(t) + \sum_{i=1}^{n_{enr}} P_{enr,i}(t) + \sum_{i=1}^{n_{sto}} P_{gen,i}(t) + \sum_{i=1}^{n_{res}} P_{res,i}(t) = \sum_{i=1}^{n_{dem}} P_{dem,i}(t) \quad (\text{III.9})$$

L'équation III.8 correspond au coût total de fonctionnement, qui est la somme des coûts c de chaque composant du réseau, dont les puissances sont notées avec les indices *gen* (générateurs), *enr* (énergies renouvelables), *sto* (stockage), *res* (réseau) et *dem* (demande). Quant à III.9, elle transcrit l'équilibre entre offre et demande, qui doit être en permanence maintenu, et est donc une des contraintes principales à prendre en compte. D'autres contraintes physiques ne sont pas détaillées ici mais peuvent être consultées dans [Roch 11b].

Le réseau étant en constante évolution, ce problème est résolu en continu, toutes les minutes ou secondes, soit quasiment en temps-réel. Les équations III.8 et III.9 sont construites dynamiquement à partir des mesures reçues et leurs solutions, comportant des points de fonctionnement, sont renvoyées aux agents qui peuvent décider ou non de les appliquer.

Les émissions de gaz à effet de serre étant de plus en plus encadrées, les émissions des générateurs classiques, par exemple à gaz, peuvent être considérées comme un second objectif. Moyennant une adaptation⁴ de *MPSOM* lui permettant de prendre en compte plusieurs objectifs et de retourner un ensemble de solutions dites Pareto-optimales, ce critère peut être intégré. Il requiert toutefois de choisir une solution suivant un critère à définir par l'opérateur

4. travail en cours

du système.

Un ensemble de règles permet par ailleurs d'implémenter la stratégie de gestion définie par l'opérateur du système : conditions d'utilisation du stockage, réserves de puissance, etc.

Enfin, l'algorithme de *dispatching* peut être combiné à un système expert⁵, baptisé ASSS (*Automatic Start & Stop System*), qui décide quand et combien de sources contrôlables doivent être démarrées ou arrêtées pour minimiser encore les coûts.

3.3 Problème de la *planification*

Le problème de la *planification* est une extension du problème du dispatching. Au lieu de considérer simplement les points de fonctionnement à un temps t donné, on considère des profils sur plusieurs heures voire jours. Cela permet de déterminer quand démarrer telle ou telle source, quand il est optimal de stocker de l'énergie en prévision d'un pic de demande, quand il est préférable de délester une partie de la charge, etc. La résolution de ce problème passe d'une part par l'obtention et l'utilisation de prévisions, aussi précises que possible, de la production intermittente, de la demande et des prix d'achat et de revente d'énergie, et d'autre part sur une légère modification de l'algorithme *MPSOM*. En effet, celui-ci est adapté pour que chaque solution comporte chaque point de fonctionnement à trouver à chaque temps t de la période considérée. La solution comportera donc des variables binaires déterminant si une unité est à démarrer ou non. D'autres contraintes, telles que les durées minimales de fonctionnement ou d'arrêt des sources thermiques, sont également à ajouter.

La dimension du problème étant alors bien plus grande, il est également bien plus long à résoudre. Toutefois, il est résolu en parallèle de celui du *dispatching*, et ne nécessite pas d'être résolu en temps réel (une à deux résolutions par jour peuvent suffire, suivant la précision des prévisions).

4 Exemples d'applications

Deux configurations de réseau sont utilisées pour tester les performances de notre approche. La première est composée d'un micro-réseau où la stabilité de nos algorithmes d'optimisation sera testée et la seconde représente une centrale électrique composée de piles à combustible, où l'optimisation portera sur le coût total de la simulation. Les deux tests sont effectués sur une période de cinq jours et sont basés sur les profils de charge réel⁶. Les tests ont été effectués à

5. travail en cours

6. <http://www.sce.com/AboutSCE/Regulatory/loadprofiles>

plusieurs reprises et retournent des résultats très similaires. Côté implémentation, le système présenté dans les parties précédentes a été implémenté en Java sur la base du middleware pour SMA JADE, de façon à créer un simulateur autonome, entièrement paramétrable et adaptable à divers problèmes et structures.

4.1 Micro-réseau

Le premier exemple que nous présentons est celui d'un micro-réseau, où des piles à combustible, des panneaux photovoltaïques et des éoliennes sont installés. Une connexion au réseau de distribution est également disponible. Un profil de charge d'une durée de 5 jours est adapté à partir d'un relevé fourni par un distributeur d'électricité⁷. Notons qu'aucune modification n'a été apportée au système de gestion, à part lui indiquer les caractéristiques du nouveau micro-réseau. Les résultats, visibles sur la figure III.9⁸, montrent que le système a été capable de répondre à la demande tout au long de la simulation, malgré la forte intermittence de la production renouvelable.

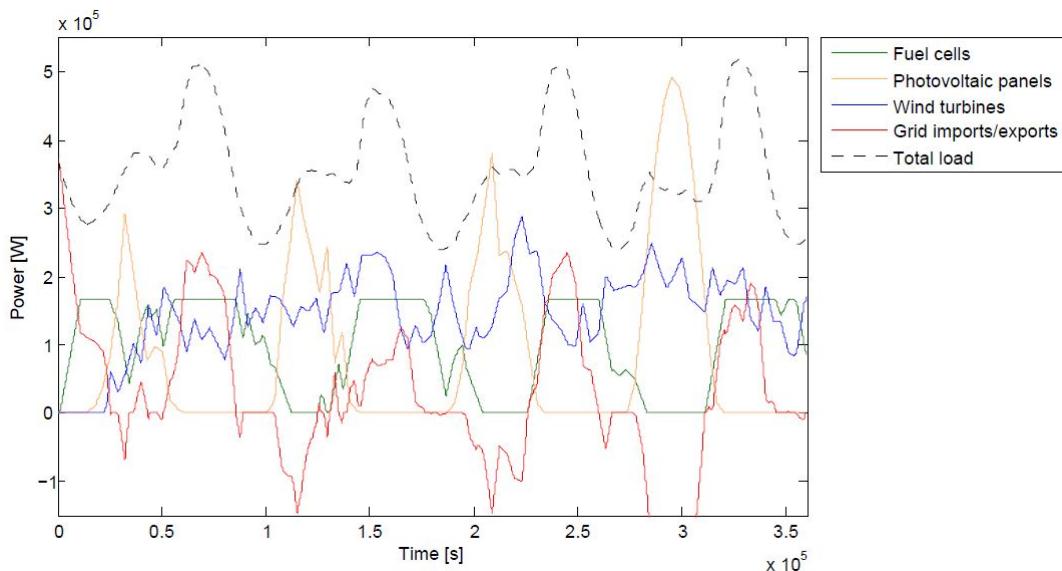


Figure III.9 – Résultats de la simulation pour le micro-réseau, montrant que le système est capable de maintenir l'équilibre production-demande même avec une forte intermittence de la production.

La tableau III.1 présente les résultats obtenus par les algorithmes *MPSOM*, *DE* et *ICA*. L'analyse de ce tableau montre que l'algorithme *MPSOM* obtient le meilleur coût suivi de l'algorithme *ICA*.

7. www.sce.com/AboutSCE/Regulatory/loadprofiles

8. Le système a recours au réseau uniquement lorsqu'il ne peut pas satisfaire la demande en utilisant les moyens d'énergies qu'il commande.

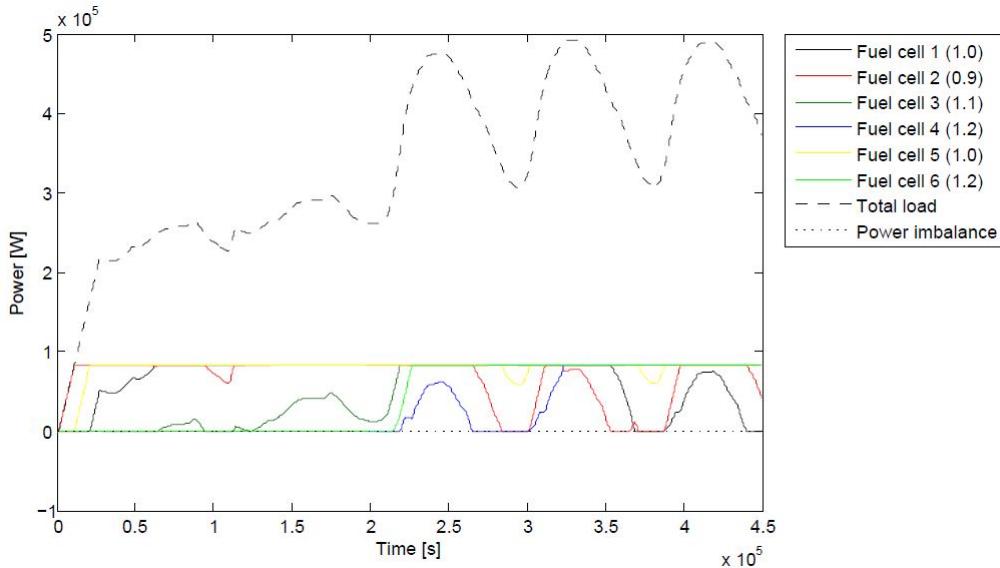


Figure III.10 – Résultats de la simulation pour le micro-réseau avec l'optimisation, ici par *ICA*. Le résultat obtenu est toutefois très poche de celui obtenu par *MPSOM*.

Algorithme	Unité	MPSOM	DE	ICA
Coût Total	€	1075	1155	1088
Déséquilibre moyen	W	-0.0175	0.1344	0.1314
Temps moyen	ms	8.126	4.075	257.0

Tableau III.1 – Comparaison des résultats pour le micro-réseau

4.2 Centrale de production

Un second exemple correspond à la gestion de l'énergie appliquée à une centrale de production pouvant comporter différents composants. Ici, on considère qu'elle comporte six piles à combustible, alimentées en hydrogène. Chaque pile possède son rendement propre, dépendant de son historique d'utilisation et de son âge. Le profil de charge utilisé est similaire au précédent⁹. Les résultats du *dispatching* de la figure III.10 montrent que l'optimisation parvient à correctement répartir l'utilisation des piles en fonction de leurs caractéristiques. On observe en effet que les piles les plus efficaces sont plus utilisées que les autres (cf. légende : une valeur plus faible indique une meilleure efficacité).

L'analyse du tableau III.2 montre les gains obtenus en utilisant trois algorithmes d'optimisation. On peut remarquer que l'algorithme ICA minimise le coût de fonctionnement à 2119 €, contre 4097 € si l'on ne fait que diviser la charge totale par le nombre de piles. Cependant, comme avec le test précédent, il est aussi plus lent que les autres algorithmes. Differential

9. l'Ensoleillement et la vitesse du vent sont extraits de : www.unige.ch/cuepe/html/meteo/donnees-csv.php

Algorithme	Unité	EqD	MPSOM	DE	ICA
Coût Total	€	4097	2165	2160	2119
Déséquilibre moyen	W	≈ 0	-0.099	-2.286	-2.888
Temps moyen	ms	< 1	11.82	8.391	454.8

Tableau III.2 – Comparaison des résultats pour la centrale de production

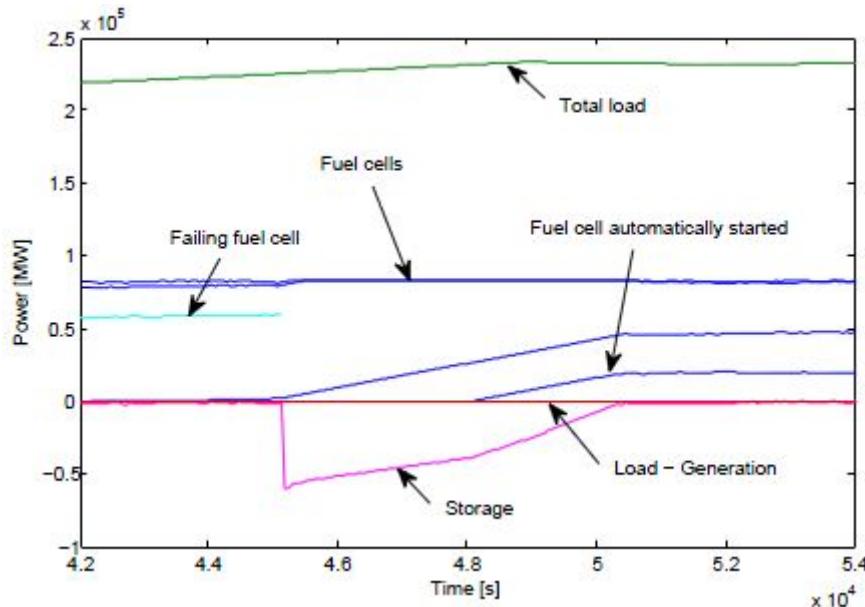


Figure III.11 – Dans cette figure, la puissance est achetée sur le réseau afin de compenser la défaillance de la batterie. La puissance achetée via le réseau est comptée comme négatif.

Evolution (DE) est particulièrement rapide, mais est également assez imprécise, bien que le déséquilibre ne représente que 10^{-4} % de la charge totale maximale. Il convient également de noter que le déséquilibre de *MPSOM* est presque toujours proche de zéro. Par conséquent, il sera le plus privilégié.

D'autres tests montrent que le système est capable de réagir correctement lorsqu'une panne survient, en utilisant au mieux les possibilités offertes par les différents éléments du réseau. La figure III.11 illustre la réaction du système face à la défaillance d'une batterie.

5 Conclusion

Les travaux présentés dans ce chapitre ont montré qu'un système de gestion d'énergie à la fois flexible et efficace a été développé, en combinant les avantages des systèmes multi-agents et des métahéuristiques. De nombreuses applications sont possibles, et plusieurs sont en cours d'étude : gestion de centrales à gaz avec prise en compte des émissions, *trading* entre micro-

Chapitre III. Gestion intelligente de l'énergie dans les smart grids

réseaux, etc. Les perspectives incluent la prise en compte de nouveaux éléments (véhicules électriques, utilisateurs, gestion de la demande, pertes en ligne, etc.) ainsi que la simulation en temps réel des réseaux interconnectés.

Chapitre IV

Métaheuristiques pour la réduction de l'énergie consommée en mémoire dans les systèmes embarqués

Les travaux décrits dans ce chapitre ont donné lieu aux publications suivantes : [[Idou 10e](#)], [[Idou 10c](#)], [[Idou 10f](#)], [[Idou 10d](#)], [[Idou 11b](#)].

Nos deux articles joints à ce mémoire sont :

- **L. Idoumghar**, M. Idrissi-Aouad, M. Melkemi and R. Schott, “Hybrid PSO-SA Type Algorithms for Multi-Modal Function Optimization and Reducing Energy Consumption in Embedded Systems”. Journal of Applied Computational Intelligence and Soft Computing, vol. 2011, pp. 1-12 , juin 2011.
- **L. Idoumghar**, M.Idrissi-Aouad, M. Melkemi and R. Schott, “Metropolis Particle Swarm Optimization Algorithm with Mutation Operator For Global Optimization Problems”. 22th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2010), pp. 35-42, October 27-29, 2010, Arras, France.

Dans ce chapitre nous présentons quelques métáheuristiques hybrides que nous avons mises en œuvre pour réduire l'énergie consommée en mémoire dans les systèmes embarqués. Ce chapitre est une synthèse des travaux de thèse de Mme Idrissi-Aouad [[Idri 11](#)] que j'ai co-encadrée (à 33%), avec le Professeur R. Schott (34%) et O. Zendra (33%), et qui a été soutenue en juillet 2011.

1 Introduction

Les systèmes embarqués font partie intégrante de notre vie quotidienne : ordinateur, smartphone, etc. Avec l'évolution de la technologie, ces systèmes intègrent de plus en plus de fonc-

tionalités complexes (Internet, vidéophonie, vidéo, etc.) nécessitant une mémoire de plus en plus grande et donc une batterie plus grande. Par conséquent, la mémoire deviendra l'élément le plus consommateur d'énergie dans un système embarqué. En effet, les tendances dans [ITRS 07] montrent que la consommation des systèmes intégrés (*SoC*) est dominée par la consommation mémoire dynamique et statique (voir Figure IV.1). De ce fait, en 2022, la consommation mémoire représentera environ 75.4% de la consommation totale d'un système intégré [ITRS 07]. Se pose alors la problème de la réduction de l'énergie consommée en mémoire. Dans ce chapitre, nous présentons quelques travaux¹ menés dans le cadre du projet ANR² MORE³. L'objectif de ce projet, conjoint entre le LORIA, l'IRIT⁴ et le LIP6⁵, est de définir des stratégies de recherche de compromis entre trois critères : consommation énergétique, occupation mémoire et temps d'exécution pire-cas (WCET⁶).

Le LORIA devait se focaliser sur l'aspect énergie, un des trois critères du projet MORE. Cela consistait à attaquer le problème de l'utilisation énergétique dans les systèmes embarqués temps-réel d'un point de vue logiciel, en travaillant sur la gestion mémoire et en considérant l'architecture matérielle sous-jacente.

Ces différentes approches peuvent être divisées en deux catégories : optimisations matérielles et optimisations logicielles. Les techniques d'optimisations matérielles, hors du contexte de notre travail, ne seront pas abordées dans ce chapitre, mais une quantité importante de littérature les concernant est disponible (voir les premières parties de [Gray 02]). Dans la suite de ce chapitre, nous nous focaliserons sur les techniques logicielles assistées par le compilateur. Dans ce cadre, les *Scratch-Pad Memories (SPMs)* ou mémoires *Scratch-Pad* présentent un intérêt considérable de par la facilité et la certitude avec lesquelles leur comportement (temps de réponse notamment) peut être prédit. Il faudra donc les intégrer dans notre plate-forme.

2 Problématique

2.1 Composants de la hiérarchie mémoire

Dans cette section, nous présenterons, brièvement, les différents comportements énergétiques des principaux composants de la hiérarchie mémoire. En effet, une hiérarchie mémoire se

1. Travaux de thèse de Mme Idrissi-Aouad
2. ANR : Agence Nationale de la Recherche.
3. MORE : *Multicriteria Optimization for Real-time Embedded systems*.
4. IRIT : Institut de Recherche en Informatique de Toulouse.
5. LIP6 : Laboratoire d'Informatique de Paris 6.
6. WCET : *Worst-Case Execution Time*.

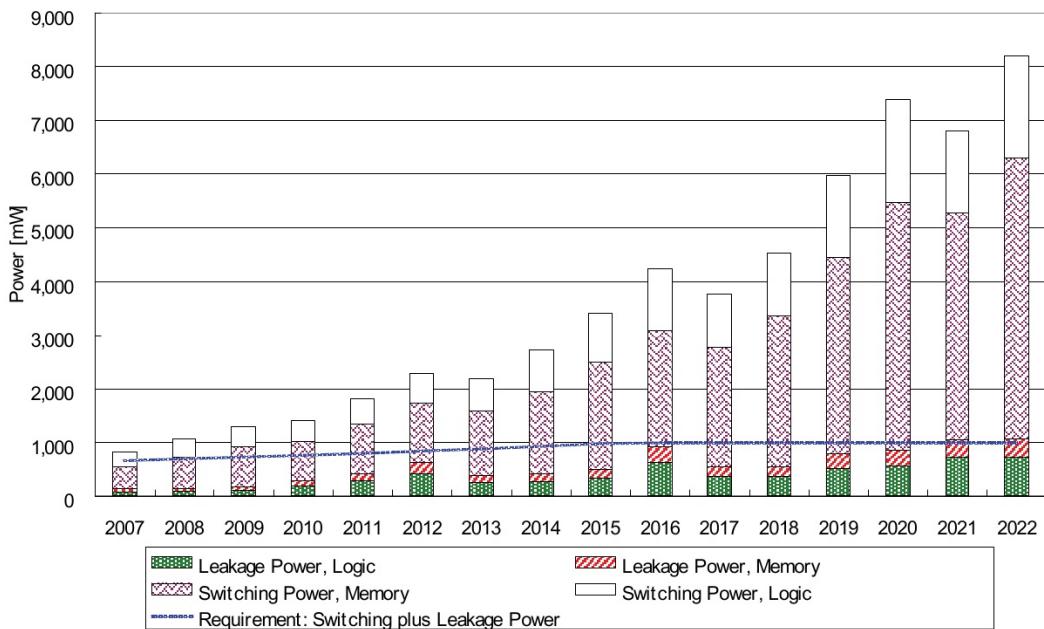


Figure IV.1 – Tendances de la consommation énergétique des SoCs [ITRS 07].

compose principalement d'une mémoire cache, d'une mémoire *Scratch-Pad* et d'une mémoire principale.

2.1.1 Mémoire Cache

La mémoire cache est une mémoire rapide et de petite taille. Elle contient les mots mémoire les plus récemment utilisés accélérant ainsi l'accès à ces mots. Lorsqu'un pourcentage important de mots nécessaires se trouve en cache, la latence effective de la mémoire est fortement réduite. L'amélioration à la fois de la bande passante et de la latence s'obtient en ayant recours à des caches multiples. Une technique de base très efficace, qualifiée de caches séparés, consiste à introduire deux caches distincts : un pour les instructions et un autre pour les données. Les opérations mémoire peuvent donc être réalisées indépendamment dans chaque cache, doublant la bande passante du système de mémoire [Tane 05]. D'après [Ben 05], la mémoire cache est une mémoire qui consomme moins d'énergie que la mémoire principale (*DRAM*⁷) mais plus que la mémoire *Scratch-Pad* (*SPM*). Selon le choix de l'utilisateur, différentes configurations possibles de mémoires cache existent ; un cache de données et un cache d'instructions, un cache de données seulement, un cache d'instructions seulement ou encore un cache unifié. Il est à noter que de nombreux processeurs contiennent une mémoire cache.

7. DRAM : *Dynamic Random Access Memory*.

Chapitre IV. Réduction de l'énergie consommée en mémoire SPM

Bien que les mémoires cache améliorent la rapidité du programme, elles ne sont pas toujours adéquates dans le cas des systèmes embarqués : elles augmentent la taille du système ainsi que son coût énergétique (surface de la mémoire cache plus sa gestion logicielle).

2.1.2 Mémoire *Scratch-Pad*

Les mémoires *Scratch-Pad* (*SPMs*) présentent un intérêt considérable de par la facilité et la certitude avec lesquelles leur comportement (temps de réponse notamment) peut être prédict. La SPM est une petite zone mémoire SRAM⁸ rapide qui ressemble à la mémoire cache mais qui est gérée au niveau logiciel, soit par le développeur, soit par le compilateur, alors que les mémoires cache nécessitent une circuiterie dédiée. Comparée à la mémoire cache, la mémoire *Scratch-Pad* possède plusieurs avantages [Zend 06, Idri 07]. En effet, la SPM consomme jusqu'à 40% moins d'énergie et 34% moins de surface que la mémoire cache [Bana 02]. De plus, le temps d'exécution d'une SPM utilisant un simple algorithme d'allocation statique de type sac à dos est 18% meilleur comparé à une mémoire cache [Bana 02]. [Ben 05] ont également montré l'efficacité d'utiliser une SPM dans une architecture mémoire où une économie de consommation de 35% est réalisée en comparaison avec une architecture mémoire sans aucune SPM. De plus, son coût est plus bas et sa gestion logicielle la rend plus prévisible, ce qui est une caractéristique importante des systèmes embarqués temps-réel.

D'après [Netw 02, Bras 02], une grande variété de puces comprenant une SPM est disponible aujourd'hui sur le marché. Les exemples de familles de processeurs embarqués contenant une SPM sont nombreux : Motorola MPC500, Analog Devices ADSP-21XX, Philips LPC2290, Atmel AT91-C140 et tant d'autres [Nguy 07]. Selon [Nguy 05], il y a au moins 80 processeurs embarqués de ce type ayant une SPM et une mémoire principale (*DRAM*) directement accédées par le CPU⁹, mais sans aucune mémoire cache. De plus, les tendances dans [LCTE 03] indiquent que la prévalence des mémoires SPM dans les systèmes embarqués est susceptible de continuer dans le futur.

D'ailleurs, plusieurs auteurs ont essayé de tirer profit des avantages des SPMs et différentes directions de recherche ont été explorées [Absa 05, Angi 03, Absa 06, Kand 02]. Ces techniques et algorithmes, synthétisés dans [Zend 06] et [Beni 00], essayent d'allouer, de façon optimale, les données et/ou le code d'une application à la SPM afin de réduire la consommation énergétique des systèmes embarqués. Une liste de références bibliographiques plus exhaustive est donnée dans [Beni 00].

8. SRAM : *Static Random Access Memory*.

9. CPU : *Central Processing Unit*.

2.1.3 Mémoire Principale

La mémoire principale (*DRAM*) est la plus grande en terme de capacité, mais c'est aussi la plus gourmande en consommation d'énergie comme c'est démontré dans [Ben 05] où les énergies consommées dans les trois mémoires cache, SPM et DRAM sont comparées. À noter également que l'accès à la DRAM est très lent.

2.2 Description de l'architecture mémoire considérée

L'architecture mémoire considérée, dans ce travail, est composée d'une mémoire *Scratch-Pad (SPM)*, d'une mémoire principale (*DRAM*) et d'un cache d'instructions. Les instructions du code source des programmes tests sont allouées en cache d'instructions alors que les données sont allouées, soit en SPM, soit en DRAM, selon la politique de l'heuristique considérée. Nous utilisons dans notre simulation une configuration mémoire spécifique qui respecte certaines contraintes :

- La taille de la mémoire cache doit être supérieure à 64 octets d'après une contrainte du projet *MORE*.
- Les caractéristiques du cache de données et de la SPM doivent être les mêmes en vue de pouvoir comparer leurs performances énergétiques de façon équitable.
- Nous sommes dans un cas de systèmes embarqués donc il y a certaines valeurs technologiques à respecter.

C'est pourquoi, nous considérons les caractéristiques présentées dans le tableau IV.1. Nous avons ajouté l'exemple d'un cache de données en vue de montrer clairement la différence entre ce dernier et la SPM.

Type de mémoire	Taille	Largeur des blocs	Associativité	Technologie
Cache d'instructions	128 Ko	16 o	1	40 nm
Mémoire DRAM DDR3	800 Ko		0	78 nm
Cache de données	1 Ko	8 o	0	40 nm
Mémoire SPM	1 Ko	8 o	0	40 nm

Tableau IV.1 – Caractéristiques des mémoires considérées.

Du fait que la SPM est gérée de façon logicielle, il n'y a aucun accès avec échec puisqu'à tout moment on connaît l'emplacement des données en SPM. Le calcul de l'énergie consommée en SPM dépend alors du coût d'un seul accès à cette mémoire ainsi que du nombre total d'accès effectués à la SPM. Le calcul de l'énergie consommée en DRAM se fait de la même manière.

Chapitre IV. Réduction de l'énergie consommée en mémoire SPM

Pour notre architecture mémoire composée d'une SPM, d'un cache d'instruction et d'une DRAM, le modèle général de la consommation d'énergie est donné par l'équation IV.1 où les trois termes font référence à l'énergie totale consommée respectivement en SPM, en cache d'instruction et en DRAM.

$$E = E_{tspm} + E_{tic} + E_{tdram} \quad (\text{IV.1})$$

Ainsi, nous obtenons finalement la formule détaillée ci-dessous :

$$E = N_{spmr} * E_{spmr} \quad (\text{IV.2})$$

$$+ N_{spmw} * E_{spmw} \quad (\text{IV.3})$$

$$+ \sum_{k=1}^{N_{icr}} [h_{ik} * E_{icr} + (1 - h_{ik}) * [E_{dramr} + E_{icw} \\ + (1 - WP_i) * DB_{ik} * (E_{icr} + E_{dramw})]] \quad (\text{IV.4})$$

$$+ \sum_{k=1}^{N_{icw}} [WP_i * E_{dramw} + h_{ik} * E_{icw} + (1 - WP_i) * \\ (1 - h_{ik}) * [E_{icw} + DB_{ik} * (E_{icr} + E_{dramw})]] \quad (\text{IV.5})$$

$$+ N_{dramr} * E_{dramr} \quad (\text{IV.6})$$

$$+ N_{dramw} * E_{dramw} \quad (\text{IV.7})$$

Les lignes (IV.2) et (IV.3) représentent respectivement l'énergie totale consommée lors d'une lecture et lors d'une écriture en SPM. Les lignes (IV.4) et (IV.5) représentent respectivement les énergies totales consommées lors d'une lecture et lors d'une écriture en cache d'instructions. Alors que les lignes (IV.6) et (IV.7) représentent respectivement l'énergie totale consommée lors d'une lecture et lors d'une écriture en DRAM. Les différents termes utilisés ci-dessus sont expliqués dans le tableau IV.2.

L'objectif de notre travail est de minimiser la formule de l'estimation de la consommation d'énergie de notre architecture mémoire. Dans cette architecture mémoire considérée, le cache d'instruction ne sert qu'à stocker les instructions de l'application et ne peut donc pas être utilisé au stockage des données de l'application. Et étant donné les avantages des deux mémoires restantes (SPM et DRAM), il est préférable de stocker le maximum de données possibles en SPM. Dans d'autres termes, dans la formule énergétique de la ligne (IV.2) à la ligne (IV.7), les termes N_{spmr} et N_{spmw} doivent être maximisés autant que possible. Ainsi, notre problème revient de maximiser les nombres d'accès à la SPM. Cela est expliqué dans la section qui suit.

Terme	Signification
E_{spmr}	Énergie consommée lors d'une lecture en SPM.
E_{spmw}	Énergie consommée lors d'une écriture en SPM.
N_{spmr}	Nombre d'accès en lecture à la SPM.
N_{spmw}	Nombre d'accès en écriture à la SPM.
E_{icr}	Énergie consommée lors d'une lecture en cache d'instructions.
E_{icw}	Énergie consommée lors d'une écriture en cache d'instructions.
N_{icr}	Nombre d'accès en lecture au cache d'instructions.
N_{icw}	Nombre d'accès en écriture au cache d'instructions. Généralement, comme on ne modifie pas les instructions, ce terme est souvent à zéro.
E_{dramr}	Énergie consommée lors d'une lecture en DRAM.
E_{dramw}	Énergie consommée lors d'une écriture en DRAM.
N_{dramr}	Nombre d'accès directs en lecture à la DRAM.
N_{dramw}	Nombre d'accès directs en écriture à la DRAM.
WP_i	Politique d'écriture (<i>Write Policy</i>) du cache considérée (<i>write-through/write-back</i>). Dans le cas de la politique d'écriture simultanée (<i>write-through</i>), $WP_i = 1$. Dans le cas de la politique d'écriture différée (<i>write-back</i>), $WP_i = 0$.
DB_{i_k}	<i>Dirty Bit</i> utilisé dans le cas de la politique d'écriture différée pour indiquer lors de l'accès k si la ligne du cache d'instructions avait été modifiée auparavant ($DB_i = 1$) ou pas ($DB_i = 0$).
h_{i_k}	Type de l'accès k au cache d'instructions. Dans le cas d'un accès avec succès au cache d'instructions, $h_{i_k} = 1$. Dans le cas d'un accès avec échec au cache d'instructions, $h_{i_k} = 0$.

Tableau IV.2 – Liste des termes du modèle d'estimation de la consommation énergétique.

2.3 Formalisation du problème

Le problème étudié relève de l'optimisation combinatoire. Vu de plus prêt, on constate que c'est un problème de type sac à dos (*knapsack*) [H Ke 04].

Nous voulons remplir la mémoire *Scratch-Pad*, qui peut contenir une capacité maximale de C , avec une combinaison de données à partir d'une liste de N données possibles. Chaque donnée est caractérisée par sa taille $size_i$ et par le nombre de fois que l'on y accède : son nombre d'accès an_i . Soit s une solution au problème. s est alors une suite finie de N termes telle que $s[n]$ est soit 0, soit la taille de la $n_{i\text{eme}}$ donnée. $s[n] = 0$ si et seulement si la $n_{i\text{eme}}$ donnée n'est pas sélectionnée dans l'espace de la solution. Le but est de maximiser la somme des nombres d'accès des données allouées en SPM sans dépasser la capacité maximale C de la SPM. Cette

fonction objectif est exprimée par l'équation IV.8.

$$\begin{cases} \text{Maximiser} & \sum_{i=1}^N a n_i \frac{s[i]}{\text{size}_i} \\ \text{Tel que :} & \sum_{i=1}^N s[i] \leq C \end{cases} \quad (\text{IV.8})$$

3 Programmes tests utilisés

Les expérimentations concernant l'optimisation de la consommation d'énergie en mémoire sont réalisées avec un ensemble de programmes tests sélectionnés dans le cadre du projet *MORE*. Ces programmes tests sont issus de six suites différentes : *MiBench* [Guth 01], *SNU-RT*, *Mälardalen*, *Mediabenchs*, *Spec 2000* et *Wcet Benchs*. Le tableau IV.3 donne une description de ces programmes tests. Ils peuvent également être téléchargés à partir de [Benc 10].

Programmes tests	Suite	Description
ShaCE	MiBench	The secure hash algorithm that produces a 160-bit message digest for a given input.
BitcountCE	MiBench	Tests the bit manipulation abilities of a processor by counting the number of bits in an array of integers.
FirCE	SNU-RT	Finite impulse response filter (signal processing algorithms) over a 700 items long sample.
JfdctintCE	SNU-RT	Discrete-cosine transformation on 8x8 pixel block.
AdpcmCE	Mälardalen	Adaptive pulse code modulation algorithm.
CntCE	Mälardalen	Counts non-negative numbers in a matrix.
CompressCE	Mälardalen	Data compression using lzw.
DjpegCE	Mediabenchs	JPEG decoding.
GzipCE	Spec 2000	Compression.
NsichneuCE	Wcet Benchs	Simulate an extended Petri net. Automatically generated code with more than 250 if-statements.
StatemateCE	Wcet Benchs	Automatically generated code.

Tableau IV.3 – Liste des programmes tests utilisés.

4 Algorithmes de résolution

4.1 Synthèse des heuristiques classiques utilisées

Dans tous les travaux présentés dans les sections précédentes, le placement des données est, comme nous l'avons vu, guidé par les caractéristiques de la mémoire considérée (rapidité d'accès, coût énergétique, grand nombre de cas des accès avec échec, etc.). À cause de la taille réduite de la mémoire SRAM, il s'agit d'y allouer les données de façon optimale afin de réaliser des gains en énergie plus importants. Une approche est de placer les données intéressantes dans une mémoire à faible coût d'accès (*SPM*) alors que les autres données sont placées dans une mémoire à faible coût de stockage (*DRAM*). Ces données peuvent être les données les plus fréquemment accédées/utilisées [Avis 02, Domi 05, Uday 02, Wehm 04] ou les données les plus conflictuelles dans le cache [Pand 97, Truo 98].

En vue de déterminer ces données intéressantes, ces méthodes utilisent des données de profils (*data profiling*) dans le but d'obtenir des informations sur la fréquence des accès mémoire. Ces informations peuvent être collectées soit statiquement en analysant le code source des programmes tests, soit dynamiquement, à travers les profils d'exécution des programmes (nombre de fois qu'une donnée est accédée, taille des données, fréquence d'accès, etc.). Ainsi, au lieu d'utiliser une heuristique gloutonne, la plupart des auteurs utilisent l'une des quatre heuristiques suivantes :

1. **Allouer les données en mémoire SRAM selon leur taille** : les données les plus petites sont allouées en SRAM tant qu'il y a de la place disponible, sinon elles sont allouées en mémoire principale DRAM. Cette méthode a l'avantage d'être simple à implanter puisqu'elle ne tient compte que de la taille des données, mais elle a pour inconvénient d'allouer les données les plus grandes en DRAM sachant que si ces données sont souvent accédées, il en résultera très peu d'économie d'énergie.
2. **Allouer les données en mémoire SRAM selon leur nombre d'accès** : les données les plus souvent accédées/utilisées sont allouées en SRAM tant qu'il y a de la place disponible, sinon elles sont allouées en mémoire principale. Cette transformation est déjà plus optimale que la première puisque les données les plus souvent accédées/utilisées seront allouées dans une mémoire qui consomme moins d'énergie. Elle permettra donc de réaliser plus d'économies comme cela est expliqué et démontré dans [Stei 02]. Cependant, nous pouvons noter des problèmes de granularité dans certains cas. Citons comme exemple, une structure dont seule une partie est très souvent accédée/utilisée.
3. **Allouer les données en mémoire SRAM selon leur nombre d'accès et leur**

taille (nombre/octet) : c'est en quelque sorte une combinaison des deux heuristiques précédentes. L'idée est de combiner leurs avantages. Si l'on considère l'exemple d'une structure dont seule une partie est le plus souvent accédée/utilisée, on tient compte du nombre moyen d'accès à cette structure. On évite ainsi les problèmes de granularité. Dans cette heuristique, appelée *BEH*¹⁰, les données sont triées en fonction de leurs ratios (nombre d'accès/taille) dans l'ordre descendant. La donnée ayant le ratio le plus élevé est allouée la première en SRAM tant qu'il y a de la place disponible. Autrement, elle est allouée en mémoire principale. Cette heuristique a l'inconvénient d'utiliser une méthode de tri qui peut être lourde et coûteuse en temps de calcul pour une quantité importante de données. De plus, cette heuristique ne fonctionnera pas très bien dans une perspective dynamique où la capacité maximale de la mémoire SRAM n'est pas connue à l'avance.

4. **Allouer les données en mémoire SPM en utilisant la méthode Tabou** : pour pallier aux inconvénients de la méthode *BEH* un algorithme de résolution basé sur la méthode tabou a été proposé par [Idri 10]. Les auteurs de ce travail ont montré qu'ils obtiennent les mêmes résultats que *BEH* comme le montre la figure IV.2. Dans la suite, nous nous référons à la méthode *TS* comme base de comparaison de nos optimisations de la consommation d'énergie.

4.2 Métaheuristiques hybrides

4.2.1 Algorithme génétique hybride

Dans cette sous-section, nous présentons un algorithme hybride, proposé dans [Idou 10e] et noté *GASA*, qui est basé sur le principe des algorithmes génétiques et de la méthode du recuit simulé. Dans cet algorithme, l'algorithme génétique utilise le recuit simulé comme un troisième opérateur, appelé *opérateur d'amélioration*. Ses principes sont décrits dans l'algorithme 10. Les autres opérateurs sont définis comme suit :

- **Solution (*individu*)** : une solution peut être représentée par un tableau de taille égale au nombre de données. Chaque élément de ce tableau indique si une donnée est incluse dans la SPM ('1') ou pas ('0').
- **Sous-population initiale** : la population initiale P_0 est choisie aléatoirement. Dans l'algorithme 10, $PopSize = 30$ est la taille de chaque population P_0 . Durant chacune des $maxGen$ générations, les résultats de $PopSize$ sont générés à travers le croisement de parents sélectionnés à partir de la sous-population.

10. BEH : *Best known Existing Heuristic*.

IV.4 Algorithmes de résolution

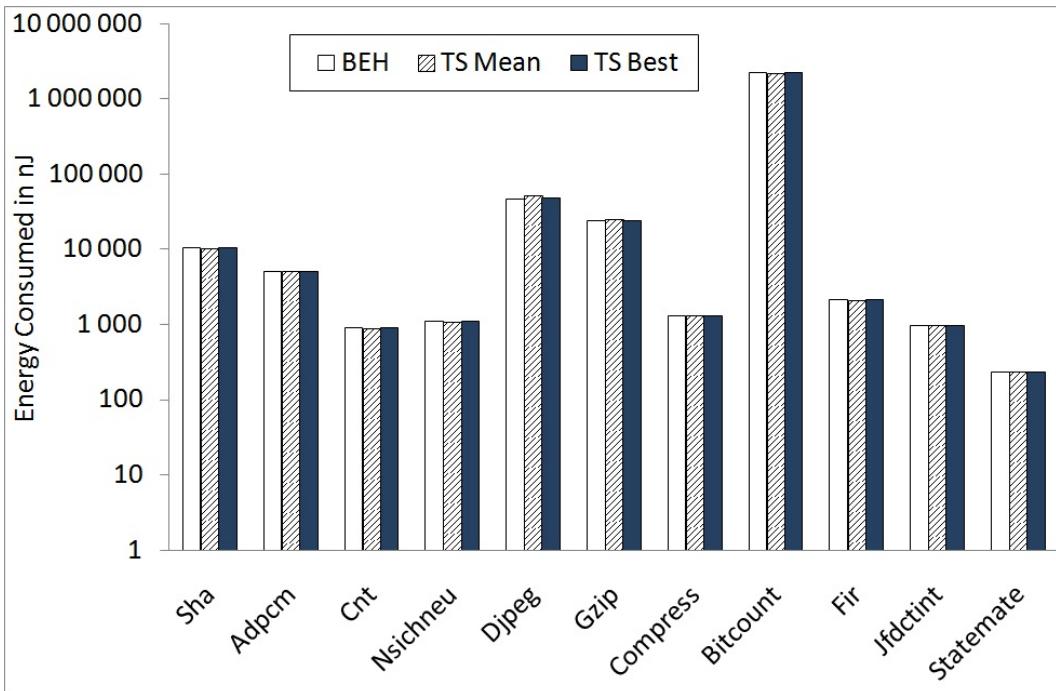


Figure IV.2 – Comparaison des énergies consommées par l’algorithme TS [Idri 10] et par *BEH*.

– **Fonction objectif** : la fonction *fitness* est définie dans l’équation IV.9 suivante :

$$Fitness(solution) = Total_Number_Access_all_data - Number_Access(solution). \quad (\text{IV.9})$$

- **Opérateur de sélection** : nous avons utilisé une sélection basée sur le principe de la roulette.
- **Opérateur de croisement** : nous avons utilisé un croisement en deux points et il est appliqué avec une probabilité $p_c = 0.6$
- **Opérateur de mutation** : nous avons utilisé une mutation binaire et elle est appliquée avec une probabilité $p_m = 0.06$.
- **Opérateur d’amélioration** : avec la probabilité $P_i = 0.01$, l’algorithme SA est appliqué au nouveau résultat.

La figure IV.3 donne les résultats moyens obtenus sur 30 exécutions indépendantes de TS [Idri 10] et 30 exécutions indépendantes de l’algorithme hybride *GASA*. *GASA* trouve toujours la meilleure solution et dépasse les performances énergétiques de TS. En effet, *GASA* consomme de 76.23% (StatemateCE) jusqu’à 98.92% (ShaCE) moins d’énergie que TS.

Algorithme 10: Algorithme hybride *GASA*.

```

1 Initialiser  $p_m, p_c, p_i \in ]0,1]$  et  $i \leftarrow 1$ 
2  $maxGen \leftarrow 10000$ 
3 Générer la population  $P_0$ 
4 Evaluer  $P_0$  et trouver le meilleur individu  $\pi^*$ 
5  $\pi_{Elite} \leftarrow \pi^*$ 
6 Tant que critère d'arrêt n'est pas vérifié faire
7    $P_i \leftarrow \emptyset$ 
8   Pour  $j \leftarrow 1$  à  $PopSize/2$  Faire
9     Sélectionner deux parents  $I_1$  et  $I_2$  de  $P_{i-1}$ 
10    Enfants  $\leftarrow (p_1, p_2)$ 
11    Avec la probabilité  $p_c$ , exécuter Enfants := croisement ( $I_1, I_2$ )
12    Avec la probabilité  $p_m$ , muter les Enfants
13    Avec la probabilité  $p_i$ , améliorer les Enfants en utilisant le recuit simulé SA (avec
         $maxiter = 200$ )
14    Evaluer les Enfants et les ajouter à  $P_i$ 
15 Fin
16 Ajouter  $P_{i-1}$  à  $P_i$ 
17 Trier  $P_i$ 
18 Garder  $PopSize$  meilleurs individus dans  $P_i$ 
19 Trouver le meilleur individu  $\pi^*$  dans  $P_i$ 
20 Si  $\pi^*$  est meilleur que  $\pi_{Elite}$  Alors
21    $\pi_{Elite} \leftarrow \pi^*$ 
22 Fin
23 Si  $fitness(\pi_{Elite}) = 0$  ou  $i = maxGen$  Alors
24   critère d'arrêt est vérifié
25 Fin
26 Mettre à jour i
27 Fin

```

4.2.2 Algorithme d'optimisation par essaim particulaire hybride

Dans cette sous-section, nous présentons l'algorithme hybride proposé dans [Idou 10d] et noté *PSOSA*, qui combine les avantages de la méthode d'optimisation par essaim particulaire (qui a une forte capacité de recherche globale) et la méthode du recuit simulé (qui a une forte capacité de recherche locale). Quelques travaux utilisant ce type d'hybridation pour résoudre d'autres problèmes peuvent être trouvés dans [Xia 06, Chao 06, Fang 07].

En hybridant PSO avec SA, l'algorithme est capable d'éviter les optima locaux. Toutefois, si SA est hybridé avec PSO à chaque itération, les coûts de calculs augmenteront et au même moment la rapidité de convergence de PSO pourra être affaiblie. C'est pourquoi, dans notre

IV.4 Algorithmes de résolution

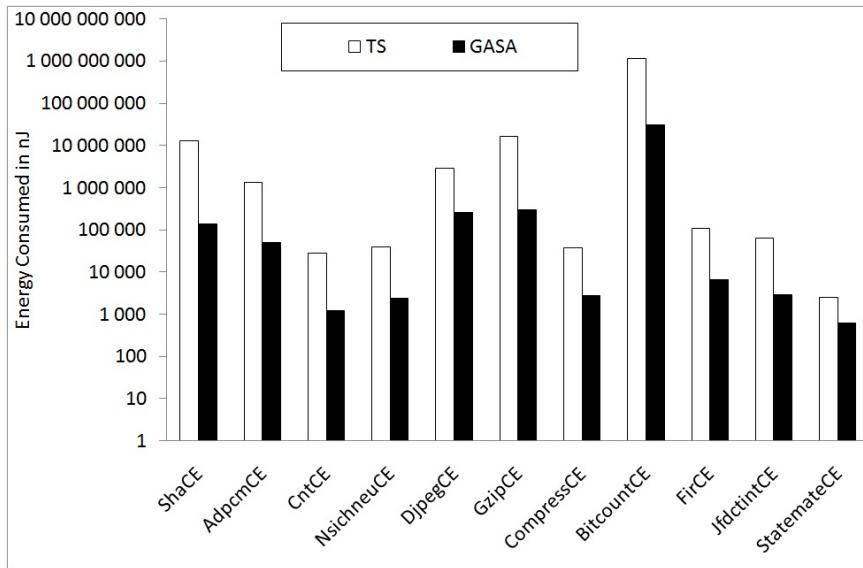


Figure IV.3 – Énergies consommées par l’algorithme hybride *GASA*.

algorithme, SA est hybridé avec PSO avec une probabilité $p_i = 0.01$ (*i.e.* comme un opérateur de mutation dans les algorithmes génétiques). Ainsi, l’algorithme *PSOSA* est en mesure de garder une convergence rapide (la plupart du temps) grâce à PSO et puis risque de sortir d’un optimum local avec l’aide de SA qui est appliqué à chaque particule courante dans l’essaim. Les principes de notre algorithme *PSOSA* sont détaillés dans l’algorithme 11, où :

- **Solution (*particule*)** : nous avons utilisé le même codage qu’un individu dans l’algorithme GASA.
- **Fonction *Evaluer*** : fonction objectif définie par l’équation IV.9.
- **Équation de mise à jour de la particule** : chaque dimension j de la particule i est mise à jour en utilisant l’équation IV.10.

$$x_{ij} = \begin{cases} 1 & \text{si } rand < \text{sigm}(v_{ij}) \\ 0 & \text{sinon} \end{cases} \quad (\text{IV.10})$$

où $\text{sigm}(v_{ij})$ est la fonction *sigmoid* utilisée pour mettre à l’échelle les vélocités comprises entre 0 et 1 et définie comme suit :

$$\text{sigm}(v_{ij}) = \frac{1}{1 + \exp(-v_{ij})} \quad (\text{IV.11})$$

- **Fonction *Générer*** : À chaque itération, l’algorithme SA génère un voisin aléatoire de la solution courante *current_solution* (ligne 25). Nous définissons la relation de voisinage

Algorithme 11: Algorithme hybride *PSOSA*.

```

1 iter  $\leftarrow 0$ , pi  $\leftarrow 0.01$ 
2 Initialiser swarm_size  $\leftarrow 30$  particules
3 maximum d'itérations maxiter  $\leftarrow 4000$ 
4 Tant que iter  $<$  maxiter faire
5   Pour toute particule i  $\leftarrow 1$  à swarm_size Faire
6     Evaluer (particule(i))
7     Si la valeur de la fitness est meilleure que la meilleure fitness (pbest)
       précédemment trouvée Alors
8       | Mettre à jour la meilleure fitness pbest
9     Fin
10   Fin
11   Choisir la particule ayant la meilleure fitness dans le voisinage (gbest)
12   Pour toute particule i  $\leftarrow 1$  à swarm_size Faire
13     Mettre à jour la vitesse en utilisant l'équation (I.4)
14     Confinement de la particule
15     Mettre à jour la position de la particule en utilisant l'équation (IV.10)
16     Confinement de la position
17   Fin
18   Si random(0, 1)  $<$  pi Alors
19     //Avec la probabilité pi, améliorer une solution current_solution en utilisant le
      recuit simulé SA
20     iterSA  $\leftarrow 0$ 
21     Initialiser T
22     best_solution  $\leftarrow current\_solution$ 
23     Ccost  $\leftarrow$  Evaluer (current_solution)
24     Tant que iterSA  $<$  400 faire
25       Neighbor  $\leftarrow$  Générer (current_solution)
26       Ncost  $\leftarrow$  Evaluer (Neighbor)
27       Si Acceptation (Ccost, Ncost, T) Alors
28         | current_solution  $\leftarrow$  Neighbor
29         | Ccost  $\leftarrow$  Ncost
30       Fin
31       Mettre à jour iterSA, best_solution
32       Mettre à jour T en utilisant l'équation (I.3)
33     Fin
34     current_solution  $\leftarrow best\_solution$ 
35   Fin
36   Mettre à jour iter
37 Fin

```

comme suit :

- avec une probabilité égale à p_i , la valeur de chaque élément de *current_solution* passe de 1 à 0 ou de 0 à 1.
- tant que *pas faisable*¹¹(*current_solution*) retirer la donnée j ayant un faible nombre d'accès de *current_solution* (*current_solution*[j] $\leftarrow 0$).

- **Fonction *Acceptation*** : cette fonction est l'idée principale dans l'approche SA. Elle spécifie la probabilité d'accepter un mouvement de *current_solution* à une solution voisine *Neighbor*, ce qui dépend également de la température (T). La fonction *Acceptation* doit satisfaire les conditions suivantes :

1. $p = 1$ si la solution *Neighbor* est meilleure que *current_solution* en termes de fonction coût f (i.e. $f(\text{Neighbor}) < f(\text{current_solution})$ dans un problème de minimisation).
2. si *Neighbor* est pire que *current_solution*, la valeur de p est positive (i.e. elle permet le déplacement vers une solution pire) mais décroît avec $|f(\text{Neighbor}) - f(\text{current_solution})|$.
3. pour *current_solution* et *Neighbor* fixes, lorsque *Neighbor* est pire que *current_solution*, la valeur de p décroît avec le temps et tend vers 0.

La fonction *Acceptation*(c_{cost}, n_{cost}, T) est décidée par la probabilité d'accepter la configuration de la solution *Neighbor*. Cette probabilité est donnée par la formule suivante :

$$p = \begin{cases} 1 & \text{si } N_{cost} < C_{cost} \\ \frac{1}{2}rand * (1 + e^{\frac{C_{cost}-N_{cost}}{T}}) & \text{sinon} \end{cases} \quad (\text{IV.12})$$

où T est la température et $rand$ un nombre aléatoire généré indépendamment dans l'intervalle $[0, 1]$.

La figure IV.4 présente les résultats moyens obtenus sur 30 exécutions de chacun des algorithmes TS[Idri 10] et PSOSA. PSOSA fait mieux que TS en termes de consommation énergétique. En effet, ces résultats montrent que PSOSA consomme de 76.23% (StatemateCE) jusqu'à 98.92% (ShaCE) moins d'énergie que TS.

11. *current_solution* devant satisfaire la contrainte de ne pas dépasser la capacité maximale de la SPM.

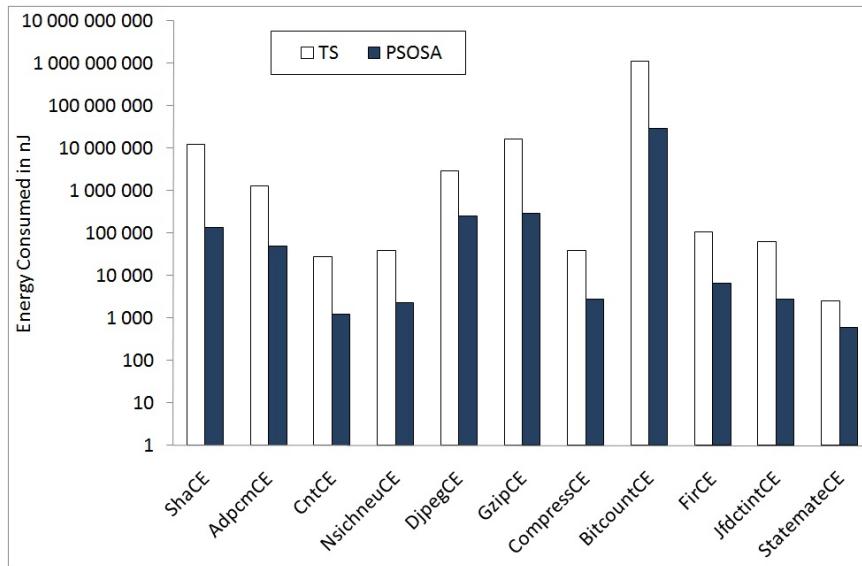


Figure IV.4 – Énergies consommées par l'algorithme hybride *PSOSA*.

4.2.3 Comparaison entre les algorithmes hybrides séquentiels

Maintenant que nous avons expérimenté différents algorithmes hybrides, il convient de les tester entre eux et cela afin de mieux se rendre compte de l'efficacité de chacun par rapport à l'autre dans le cas du problème de la réduction de la consommation d'énergie en mémoire. La figure IV.5 récapitule les résultats obtenus.

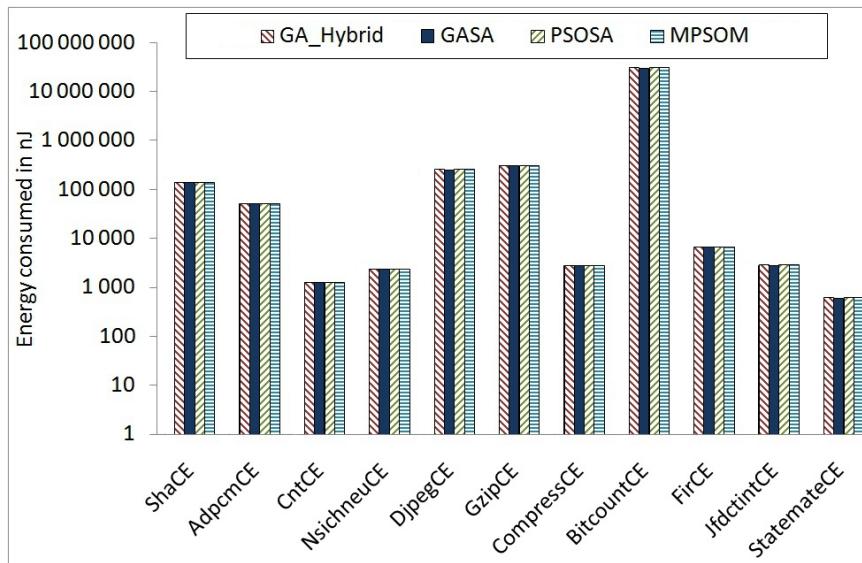


Figure IV.5 – Comparaison des énergies consommées par nos algorithmes hybrides.

Nous remarquons que tous les algorithmes hybrides atteignent les mêmes performances énergétiques que les algorithmes génétiques. Ils convergent tous vers la solution optimale.

C'est pourquoi, afin de départager les différents algorithmes, nous avons calculé les temps d'exécutions moyens obtenus sur 30 exécutions de chacun des algorithmes. La figure IV.6 donne les résultats obtenus. On peut remarquer que l'algorithme *GASA* converge rapidement en comparaison avec l'algorithme *PSOSA* et *GA_Hybrid* qui utilise la méthode tabou comme opérateur de mutation. Ceci peut s'expliquer par le fonctionnement du PSO (calcul de la vitesse) qui est un peu plus coûteux que l'algorithme GA sur ce problème.

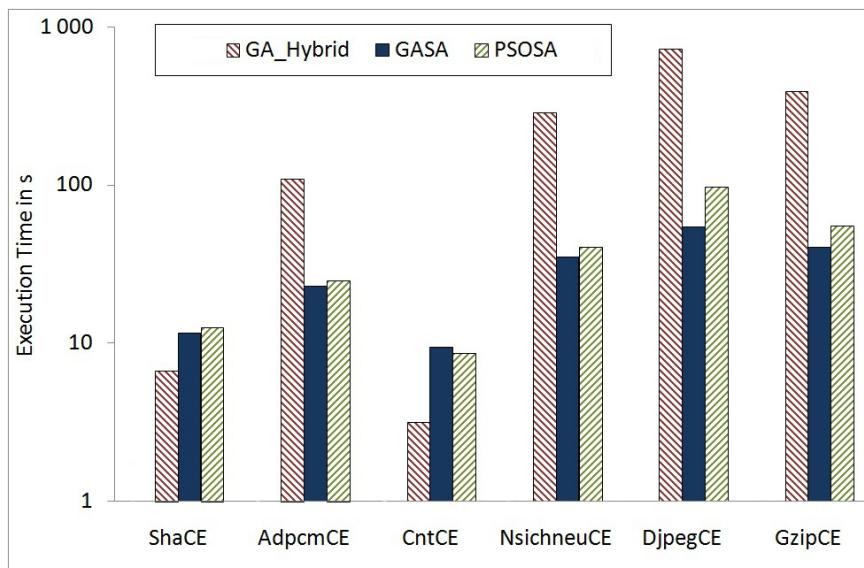


Figure IV.6 – Comparaison des temps d'exécution utilisés par nos algorithmes hybrides.

4.2.4 Algorithme GASA distribué

Dans cette sous-section, nous présentons une version distribuée de l'algorithme séquentiel hybride *GASA* décrit en sous-section 4.2.1. Cet algorithme est noté *GASA_Dist* et est proposé dans [Idou 10e]. Ses principes sont décrits dans l'algorithme 12.

Nous utilisons une architecture en îlots dont laquelle des sous-populations indépendantes évoluent séparément, sauf pour échanger un certains individus (migration). Un ensemble de $m = 30$ individus est assigné à chacun des $P = 4$ processeurs (un *cluster* de 4 PCs) pour une taille totale de la population de $m \times P$. Cet ensemble assigné à chaque processeur est sa sous-population. La sous-population initiale est choisie aléatoirement pour chaque processeur. Chaque processeur, séparément et en parallèle, exécute *GASA_Seq* (algorithme 10) sur sa

Chapitre IV. Réduction de l'énergie consommée en mémoire SPM

Algorithme 12: Algorithme coopératif et distribué *GASA_Dist* s'exécutant sur chaque processeur.

```

1 Initialiser  $p_m, p_c, p_i \in ]0, 1]$  et  $i \leftarrow 1$ 
2  $maxGen \leftarrow 2500$ 
3 Générer la population  $P_0$ 
4 Evaluer  $P_0$  et trouver le meilleur individu  $\pi^*$ 
5  $\pi_{Elite} \leftarrow \pi^*$ 
6 Tant que critère d'arrêt non vérifié faire
7    $P_i \leftarrow \emptyset$ 
8   Pour  $j \leftarrow 1$  à  $PopSize/2$  Faire
9     Sélectionner deux parents  $I_1$  et  $I_2$  de  $P_{i-1}$ 
10    Enfants  $\leftarrow (I_1, I_2)$ 
11    Avec la probabilité  $p_c$ , exécuter Enfants := croisement ( $I_1, I_2$ )
12    Avec la probabilité  $p_m$ , muter les Enfants
13    Avec la probabilité  $p_i$ , améliorer les Enfants en utilisant le récuit simulé SA (avec
         $maxiter = 200$ )
14    Evaluer les Enfants et les ajouter à  $P_i$ 
15  Fin
16  Ajouter  $P_{i-1}$  à  $P_i$ 
17  Si condition de migration est vérifiée Alors
18    | Réception de  $n = 5$  individus et les ajouter à  $P_i$ 
19  Fin
20  Trier  $P_i$  Garder les  $PopSize$  meilleurs individus dans  $P_i$ 
21  Trouver le meilleur individu  $\pi^*$  in  $P_i$ 
22  Si  $\pi^*$  est meilleur que  $\pi_{Elite}$  Alors
23    |  $\pi_{Elite} \leftarrow \pi^*$ 
24  Fin
25  Si  $fitness(\pi_{Elite}) = 0$  ou  $i = maxGen$  Alors
26    | critère d'arrêt est vérifié
27  Fin
28  Mettre à jour i
29 Fin
```

sous-population pour un certain nombre de générations. Par la suite, chaque sous-population échange un nombre spécifique d'individus (migrants) avec ses voisins; *i.e.* les migrants reçus remplaceront les mauvais individus de la sous-population. Le processus continue avec l'amélioration séparée de chaque sous-solution pour un nombre maximal de générations. À la fin, le meilleur individu existant constitue le résultat final.

Les deux versions séquentielle et distribuée produisent les mêmes résultats énergétiques pour chacune des deux heuristiques. Ces résultats sont les mêmes que ceux qui sont obtenus

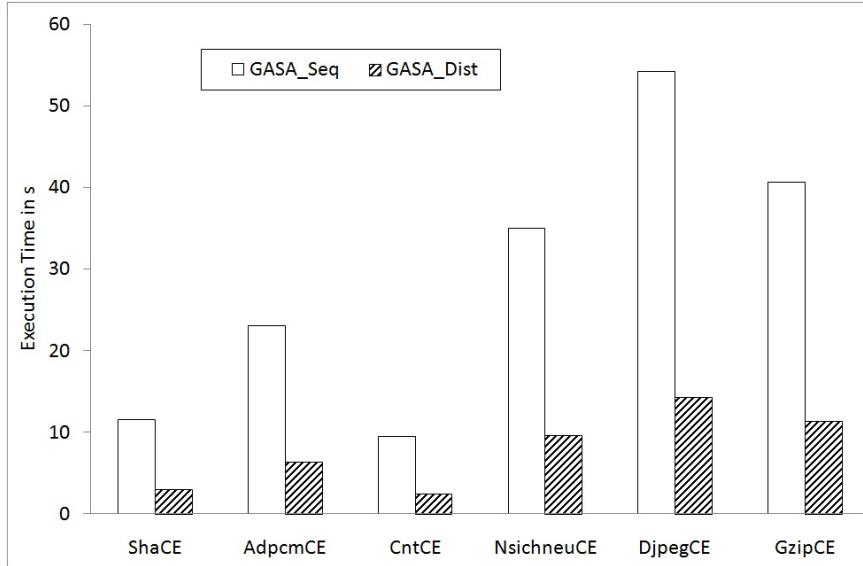


Figure IV.7 – Temps d'exécution utilisés par *GASA_Seq* et *GASA_Dist*.

auparavant dans la figure IV.3. Vu que *GASA_Seq* atteint les mêmes gains énergétiques que *GASA_Dist*, nous avons calculé les temps d'exécution de chacun d'entre eux afin de les dé-partager et cela pour les plus grands programmes tests (en termes de taille). Ainsi, la figure IV.7 donne les résultats moyens obtenus sur 30 exécutions de *GASA_Seq* et de *GASA_Dist*. Nous constatons que la version distribuée est plus rapide que la version séquentielle. En effet, *GASA_Dist* nécessite de 72.31% (GzipCE) jusqu'à 74.67% (CntCE) moins de temps d'exécution que *GASA_Seq*.

4.2.5 Algorithme PSOSA distribué

Dans cette sous-section, nous proposons une version distribuée de l'algorithme séquentiel *PSOSA* présenté en sous-section 4.2.2. Pour cet algorithme distribué, noté *PSOSA_Dist*, nous utilisons des sous-essaims indépendants de particules qui évoluent séparément, sauf pour un échange de certaines particules (migration) selon le même principe de *GASA_Dist* décrit dans la section 4.2.4.

Les deux versions séquentielle et distribuée produisent les mêmes résultats énergétiques pour chacune des deux heuristiques. Ces résultats sont les mêmes que ceux qui sont obtenus dans la figure IV.4. Comme *PSOSA_Seq* et *PSOSA_Dist* donnent les mêmes résultats énergétiques, nous avons calculé les temps d'exécution de chacun d'entre eux afin de les dé-partager et cela pour les plus grands programmes tests (en termes de taille). Ainsi, la figure IV.8 donne les résultats

moyens obtenus sur 30 exécutions de *PSOSA_Seq* et de *PSOSA_Dist*. Nous constatons que la version distribuée est plus rapide que la version séquentielle. En effet, *PSOSA_Dist* nécessite de 73.16% (AdpcmCE) jusqu'à 84.65% (CntCE) moins de temps d'exécution que *PSOSA_Seq*.

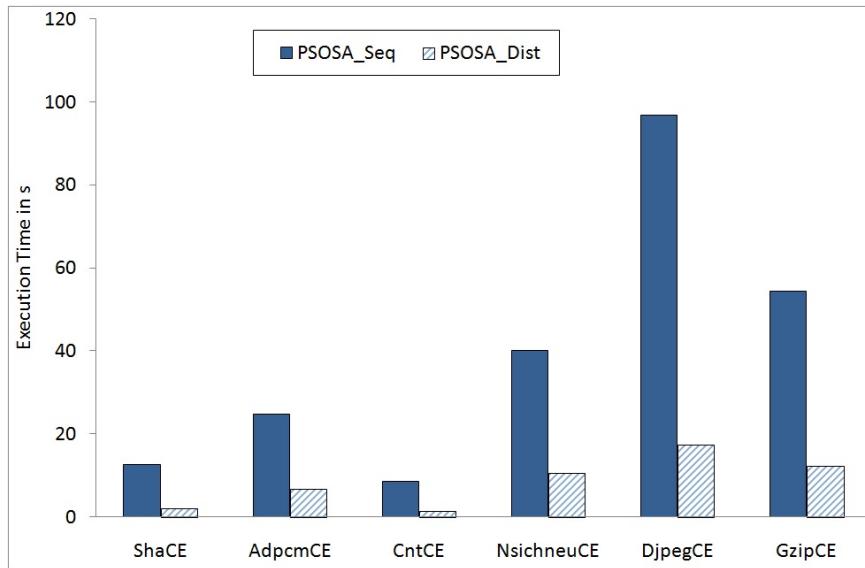


Figure IV.8 – Temps d'exécution utilisés par *PSOSA_Seq* et *PSOSA_Dist*.

5 Discussion

Nous avons présenté dans ce chapitre deux algorithmes hybrides pour réduire l'énergie consommée en mémoire dans les systèmes embarqués. Les résultats obtenus par les deux approches (en version séquentielle ou distribuée) montrent que l'on obtient la solution optimale.

L'allocation de données en mémoire est soit statique, soit dynamique. Pour l'aspect statique, nous avons proposé plusieurs algorithmes dont deux présentés dans ce chapitre. Les résultats obtenus montrent que nos algorithmes obtiennent de bons résultats vu que les solutions proposées consomment moins d'énergie en comparaison avec le meilleur algorithme proposé dans ce domaine.

Pour l'aspect dynamique, nous travaillons actuellement sur la possibilité d'utiliser des structures dynamiques telles que les *Skip-list* pour proposer un algorithme dynamique.

Chapitre V

Métaheuristiques pour l'optimisation de moteurs électriques à aimant permanent

Les travaux décrits dans ce chapitre ont donné lieu aux publications suivantes : [[Idou 10a](#)], [[Idou 11a](#)], [[Idou 09b](#)], [[Fodo 12](#)].

L'article joint à ce mémoire est :

- **L. Idoumghar**, D. Fodorean, A. N'diaye, D. Bouquain and A. Miraoui, “Simulated Annealing Algorithm for the Optimization of an Electrical Machine”. Journal of IET Electric Power Applications (accepted for publication).

Dans ce chapitre nous présentons les premiers résultats obtenus en utilisant des algorithmes d'optimisation basés sur des métahéuristiques pour optimiser les performances d'un moteur électrique. Ce travail illustre ma contribution dans deux projets internationaux :

- le projet de recherche intitulé « Intelligent Hybrid Vehicle for Individual Transportation of People with Reduced Mobility ». Projet financé par le Conseil national de la recherche de l'enseignement supérieur (acronyme en CNCSIS Roumaine) et qui est mis en œuvre par l'Université Technique de Cluj-Napoca (UTCN), Cluj, Roumanie. Le projet dirigé, par Daniel Fodorean, a débuté le 28 Juillet 2010 et pour une durée de 3 ans.
- le projet de recherche intitulé « Optimal Energy Management for Light Electric Vehicles ». Ce projet (faisant partie d'un groupe de projets financés par la Communauté Européenne, intitulé 4D-POSTDOC) et dirigé par Daniel Fodorean, a débuté le 1^{er} Juin 2010 et pour une durée de 3 ans.

Pour faciliter la compréhension des termes utilisés, nous donnons dans la section suivante quelques définitions et nous invitons le lecteur intéressé par l'expression analytique des diffé-

rentes notions décrites rapidement dans ce chapitre à télécharger l'excellent cours de Daniel Fodorean [Fodo 09].

1 Introduction

Ces dernières années, l'utilisation des Véhicules Electriques (*VE*) surtout dans l'environnement urbain offre une solution tout à fait intéressante pour le problème de la pollution atmosphérique causé en partie par l'utilisation des véhicules équipés de moteurs à combustion, sans oublier qu'il résout beaucoup de problèmes comme le bruit, la fiabilité et l'économie d'énergie. Néanmoins, les *VE* souffrent de deux inconvénients majeurs qui sont la faible autonomie et le coût de fabrication très élevé. Afin de pallier ces inconvénients, les récents travaux de recherche ([Giur 08], [hadj 08]) s'intéressent au design d'un « moteur synchrone à aimant permanent » qui constitue l'un des éléments clés d'un *VE*. L'objectif principal de ces travaux est de réduire la masse et de maximiser le couple d'un tel moteur. Ce qui permettra non seulement de minimiser le coût et la consommation, mais aussi d'augmenter considérablement l'autonomie et la fiabilité.

Structure du moteur étudié

Dans cette section nous présentons une des trois modélisations du moteur électrique que nous étudions et qui est donnée dans la figure V.1. L'objectif de l'optimisation (et enfin la fonction objectif) est de réduire la masse de la machine pour les mêmes performances données en sortie¹. Du processus de conception, fourni par Daniel Fodorean, nous avons extrait les principaux paramètres géométriques qui ont une influence importante sur les performances du moteur, voir le tableau V.1. Les caractéristiques de la machine de référence sont données dans le tableau V.2.

Définition 11

La *masse totale* m_{tot} de la partie active du moteur est calculée en utilisant la formule V.1 :

$$m_{tot} = m_{copper} + m_{stat} + m_{rot} + m_{PM} \quad (\text{V.1})$$

avec m_{copper} est la masse de cuivre placé dans les encoches statoriques, m_{stat} est la masse du stator (dents et culasse), m_{rot} est la masse du rotor et m_{PM} est la masse de l'aimant.

Définition 12

Le *couple* est une notion qui traduit la force que le moteur donne dans son mouvement. L'exemple d'un cycliste qui veut monter une côte permet de bien expliquer cette notion. Il

1. Contraintes fixées par le cahier des charges

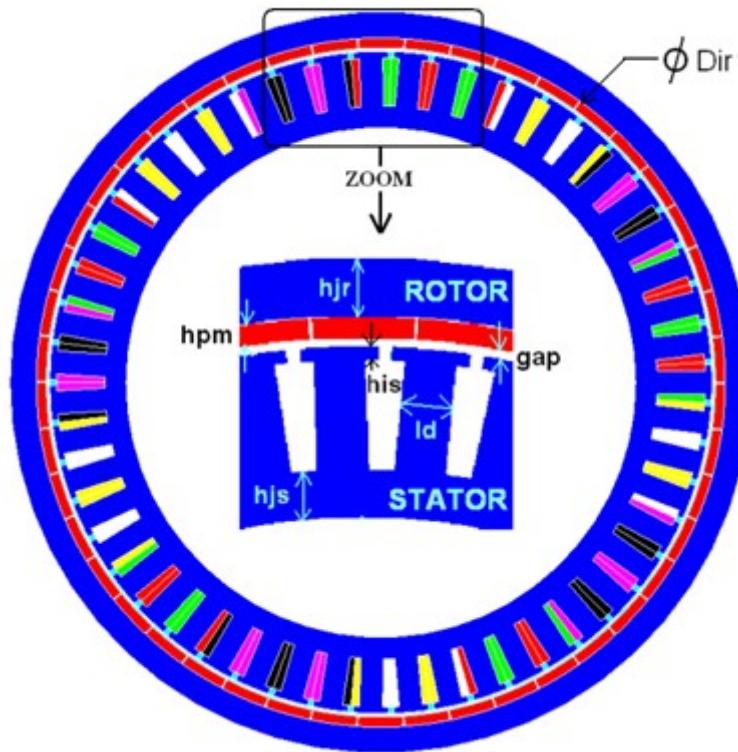


Figure V.1 – Modélisation du moteur à optimiser.

Symbol	Description	Intervalle de variation
Dir	Diamètre interne du rotor	[185 ; 210] mm
hjr	Hauteur de la culasse du rotor	[5 ; 10] mm
his	Hauteur de l'isthme de la dent	[1 ; 3] mm
hjs	Hauteur de la culasse du stator	[5 ; 10] mm
ld	Largeur de la dent	[4 ; 10] mm
gap	Longueur de l'entrefer	[0,5 ; 1,5] mm
hmp	Hauteur de l'aimant permanent	[2,5 ; 7] mm
Lm	Longueur de la machine	[30 ; 80] mm
Contraintes à respecter (C)		
T_m	Couple moteur	[21,8 ; 22,2] Nm
P_{out}	Puissance de sortie	[1490 ; 1510] W
Is	Source de courant	[15 ; 25] A

Tableau V.1 – Caractéristique de la machine à optimiser

Chapitre V. Métaheuristiques pour l'optimisation de moteurs électriques

Symbole	Description	Valeur
m_{tot}	Masse totale de la partie active	7,1207 kg
RPM	Rapport P_{out} / m_{tot}	210,67 W/kg
η	Rendement	0,9120
FP	Facteur de puissance	0,8739
c_{tot}	Coût total de la partie active	83,1590 €
P_{Fe}	Pertes fer	34,2259 W
P_J	Pertes dans le cuivre	103,2254 W

Tableau V.2 – Caractéristique de la machine de référence.

doit appuyer davantage sur les pédales pour continuer à monter la côte ; il développe ainsi davantage de couple. Le *couple d'un moteur* est calculé à l'aide de l'équation suivante :

$$T_m = \frac{P_{out}}{\Omega} \quad (\text{V.2})$$

avec :

- Ω : la vitesse de synchronisme du rotor (en rad/s) ($\Omega = 2 \times \pi \times n/60$ et $n = 3000\text{rpm}$),
- P_{out} : la *puissance utile* (appelée aussi puissance² mécanique) qui peut être déterminée par la relation suivante :

$$P_{out} = P_a - (P_{Fe} + P_J + P_m + P_s) \quad (\text{V.3})$$

avec :

- P_a : la puissance électrique consommée par le moteur,
- P_{Fe} : les pertes fer (pertes par hystérésis³, pertes par courant de Foucault⁴),
- P_J : les pertes dans cuivre (ou Joule),
- P_m : les pertes mécaniques et par ventilation font référence aux frottements des roulements et par auto-ventilation
- P_s : les pertes supplémentaires correspondent aux pertes dues aux harmoniques du champs et aux pertes dans l'entrefer⁵.

2. La puissance d'une machine est l'énergie qu'elle fournit en 1 seconde.

3. Il s'agit de la capacité d'un matériel magnétique à conserver la force magnétique au moment d'être exposée à une force de désaimantation.

4. On appelle courants de Foucault les courants électriques créés dans une masse conductrice, soit par la variation au cours du temps d'un champ magnétique extérieur traversant ce milieu (le flux du champ à travers le milieu), soit par un déplacement de cette masse dans un champ magnétique constant. Ils sont une conséquence de l'induction magnétique.

5. Il s'agit de l'espace magnétique entre les pôles d'un aimant, qui peut être rempli de n'importe quel matériel non magnétique, comme le laiton, le bois ou le plastique.

Définition 13

Le *rendement énergétique d'un moteur* correspond au rapport entre l'énergie utile fournie et l'énergie totale consommée. Son expression est donnée par la formule V.4 :

$$\eta = \frac{P_{out}}{P_a} \quad (\text{V.4})$$

Définition 14

le *facteur de puissance FP* s'apparente à un rendement : il varie de 0 à 1 et permet de quantifier, principalement le bon usage d'une ligne d'alimentation électrique. En effet une ligne est caractérisée par son aptitude à véhiculer un courant maximal. Si le facteur de puissance est proche de 1, la puissance active (seule puissance susceptible d'être transformée en énergie mécanique ou thermique) distribuée est maximale pour le courant maximal toléré par la ligne. A l'inverse si *FP* est proche de 0 la puissance active est pratiquement nulle pour le courant maximal toléré par la ligne, seules les puissances réactives et déformantes sont véhiculées. On voit tout l'intérêt, pour un distributeur d'électricité, de maintenir ce facteur de puissance proche de 1.

On peut faire une comparaison mécanique entre le facteur de puissance et le facteur d'embrayage d'une boîte de vitesses :

- lorsque la pédale d'embrayage est enfoncée, le moteur tourne (le courant circule) mais ne transmet aucune puissance au véhicule ; le facteur de puissance est nul.
- lorsque la pédale d'embrayage est relevée, le moteur tourne et toute sa quantité de mouvement est transmise au véhicule pour produire de la puissance motrice ; le facteur de puissance est unitaire.
- lorsque l'on fait patiner l'embrayage, on est dans une situation intermédiaire, cela correspond au cas où le facteur de puissance est compris entre 0 et 1.

le *facteur de puissance* du moteur est obtenu en utilisant la formule V.5 :

$$FP = \frac{P_a}{m_{mtot} \times U_{ph} \times I_s} \quad (\text{V.5})$$

avec U_{ph} est la tension par phase statorique et I_s correspond au courant total absorbé par phase statorique.

Définition 15

Le *coût total de la partie active* du moteur est égal à la somme des coûts des matériaux utilisés pour fabriquer le moteur. Il est obtenu en utilisant l'expression suivante :

$$c_{tot} = c_{aimant} + c_{rotor} + c_{stator} + c_{cuivre} \quad (\text{V.6})$$

2 Modélisation et résolution du problème

2.1 Optimisation mono-objectif

La première approche que nous avons proposée dans [Idou 11a] est basée sur une optimisation mono-objectif. La fonction objectif à optimiser est donnée par la formule V.7.

$$\text{Minimiser } f(\text{Dir}, \text{hjr}, \text{his}, \text{hjs}, \text{ld}, \text{gap}, \text{hmp}, \text{Lm}) = -RPM + \text{pénalité} \quad (\text{V.7})$$

avec :

RPM : Le rapport puissance / masse qui doit être maximisé. Il est défini par la relation suivante :

$$RPM = \frac{P_{out}}{m_{tot}} \quad (\text{V.8})$$

$$\text{pénalité} = \begin{cases} 0 & \text{Si toutes les contraintes } C \text{ sont satisfaites} \\ 10^6 \times \sum_{i=1}^3 |C_i - Limit_i| & \text{Sinon} \end{cases}$$

et,

$$Limit_i = \begin{cases} \text{Borne inférieure (i)} & \text{Si } C_i < \text{Borne inférieure(i)} \\ \text{Borne Supérieure (i)} & \text{Si } C_i > \text{Borne Supérieure (i)} \end{cases}$$

Algorithme de résolution et résultats

Pour résoudre ce problème, nous avons développé un algorithme basé sur le recuit simulé. Cet algorithme, décrit dans [Idou 11a] et joint en annexe, nous a permis d'obtenir les résultats présentés dans le tableau V.3.

L'algorithme que nous avons proposé a permis de réduire la masse de 47% et de maximiser le rapport RPM ⁶ de 88% par rapport à la modélisation de référence. C'est un très bon résultat, en comparaison avec d'autres travaux de recherche existant dans la littérature, par exemple [Giur 08] obtient un gain 32% sur la masse. En outre, puisque la masse totale de la partie active m_{tot} a sensiblement diminué, le coût total de la machine a diminué aussi. Le pourcentage de diminution du coût est plus faible que pour la masse, ce qui peut s'expliquer par une légère augmentation de la hauteur de l'aimant permanent hmp (de 3 à 5 mm) par rapport à la machine de référence.

Pour les performances énergétiques du moteur optimisé, on peut dire que le rendement et le

6. Pour qu'un moteur soit performant, RPM doit être maximal

V.2 Modélisation et résolution du problème

Symbol	Après optimisation	Gain (%)
m_{tot}	3,7697 kg	+ 47,0599
RPM	397,93 W/kg	+ 88,8878
η	0,9014	- 1,1622
P_{Fe}	37,7416 W	- 9,3151
P_J	118,9112 W	- 13,1911
FP	0,8230	- 5,8244
c_{tot}	59,3902 €	+ 28,5823
Dir	210 mm	
hjr	5 mm	
his	1 mm	
hjs	5 mm	
ld	5 mm	
gap	0.5 mm	
hmp	5 mm	
Lm	31.6 mm	

Tableau V.3 – Caractéristique de la machine optimisée.

facteur de puissance ont pratiquement conservé leurs valeurs initiales - le facteur de puissance peut être corrigé par une opération de contrôle appropriée⁷. On peut aussi remarquer que la solution optimisée a plus de pertes fer. Cette augmentation des pertes fer est due à la limitation de notre modèle de conception (et de tous les modèles d'analyse existants dans la littérature) qui ne peut pas calculer avec exactitude la répartition de la *densité de flux*⁸ en chaque millimètre du stator. Un calcul exact de la densité de flux dans toutes les parties du stator peut être obtenu avec une méthode d'analyse par éléments finis⁹. Ce qui permettra de calculer exactement les pertes fer au détriment du temps de calcul. J'invite le lecteur intéressé à consulter l'article [Idou 11a] ci-joint pour plus d'explications.

2.2 Optimisation multiobjectif

Dans cette section, nous présentons la modélisation du problème d'optimisation d'un moteur électrique à aimant permanent sous forme de problème d'optimisation multiobjectif. En effet, à partir de la modélisation mono-objectif, on rajoute un second objectif qui aura pour but de maximiser le rendement η du moteur.

7. Opération non décrite dans ce mémoire vu qu'elle nécessite des notions qui sont hors de mon champ de compétences.

8. Flux magnétique par secteur d'unité d'une section normale à la direction de flux. Aussi connu comme induction magnétique.

9. Nous travaillons sur l'intégration de l'analyse par éléments finis dans notre algorithme d'optimisation.

Notre problème d'optimisation multiobjectif consiste donc à trouver une solution X qui optimise simultanément les deux objectifs RPM et η tout en respectant les contraintes données dans le tableau V.1. Ce problème d'optimisation est défini sous la forme :

$$\left\{ \begin{array}{l} \text{Maximiser } F(X) = \{RPM(X), \eta(X)\} \\ \text{avec :} \\ 21,8 \text{ Nm} \leq T_m(X) \leq 22,2 \text{ Nm} \\ 1490 \text{ W} \leq P_{out}(X) \leq 1510 \text{ W} \\ 15 \text{ A} \leq I_s(X) \leq 25 \text{ A} \end{array} \right. \quad (\text{V.9})$$

Le problème décrit par la formule V.9 indique qu'il s'agit d'un problème de maximisation puisque le but est de maximiser à la fois le rendement η et le rapport RPM .

Algorithmes de résolution et résultats

Pour résoudre le problème V.9, nous avons utilisé trois algorithmes d'optimisation multiobjectif :

- Hybrid Archive-based Micro Genetic Algorithm (AMGA) [Tiwa 09] ;
- Archive-based Stochastic Ranking Evolutionary Algorithm (ASREA) ;
- Omni-Optimizer : A Generic Evolutionary Algorithm (OMNIOPT) ;

Les trois algorithmes sont des variantes multiobjectif de l'algorithme génétique. Ils utilisent “simulated binary crossover” [Deb 95] comme opérateur de croisement et une mutation polynomiale (polynomial mutation) [Deb 96].

Pour nos simulations, nous avons fixé les paramètres suivants pour les 3 algorithmes :

- Taille de la population : 100
- Nombre d'itérations : 1100
- Taille de l'archive finale : 100

Les résultats obtenus sont illustrés dans la figure V.2. L'analyse de cette figure montre que l'algorithme *OMNIOPT* obtient de bons résultats en comparaison avec ceux obtenus par l'algorithme *AMGA* et *ASREA*. Une comparaison des meilleurs résultats¹⁰ obtenus par les trois algorithmes est donnée dans le tableau V.4.

L'analyse de ce tableau montre que :

- Les trois algorithmes améliorent la qualité de la solution de référence avec une réduction de la masse du moteur allant de 17,218% (ASREA) à 45,718% (OMNIOPT).

10. Solutions indiquées par leurs valeurs sur le graphique de la figure V.2.

V.2 Modélisation et résolution du problème

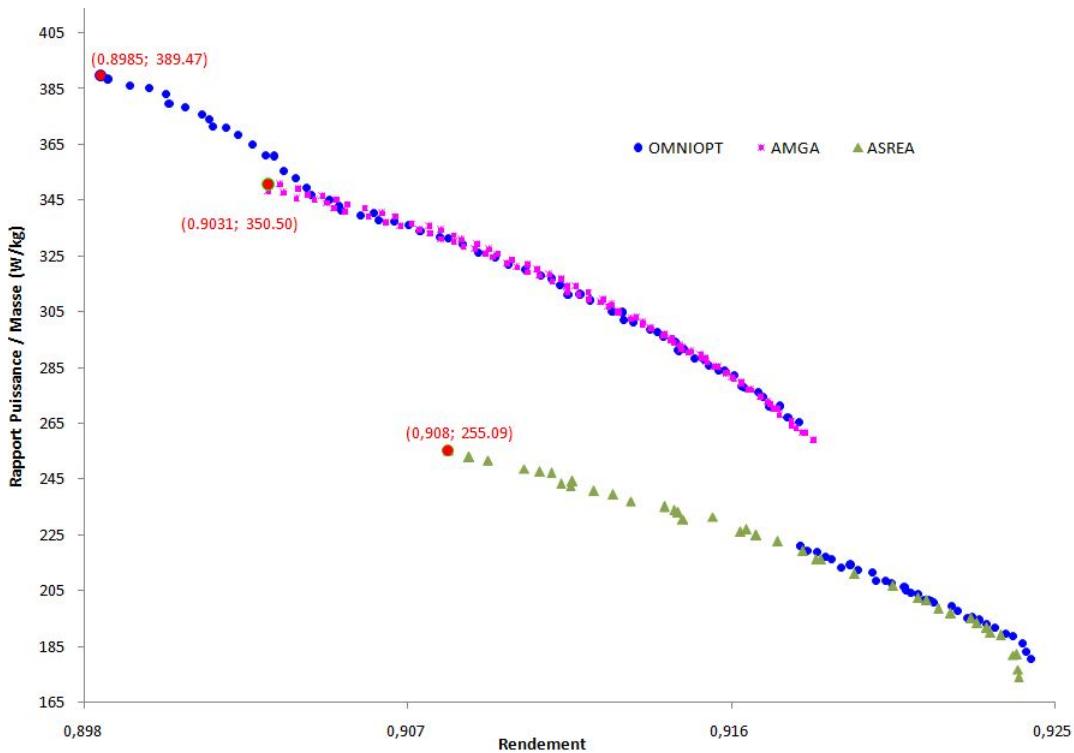


Figure V.2 – Rapport RPM en fonction du rendement.

Symbole	ASREA	AMGA	OMNIOPT
$m_{tot}(\text{kg})$	5,8946 + 17,218 Gain (%)	4,2863 + 39,805 Gain (%)	3,8652 + 45,718 Gain (%)
RPM	255,087 W/kg + 21,08 Gain (%)	347,904 W/kg + 65,14 Gain (%)	389,47 W/kg + 84,87 Gain (%)
η	0,908	0,903	0,9
FP	0,796	0,794	0,807
T_m	34,1878	33,9057	34,2273
P_{out}	1503	1491,25	1505,4
Dir	187,95 mm	201,14 mm	209,79 mm
hjr	5 mm	5 mm	5 mm
his	1 mm	1 mm	1 mm
hjs	5 mm	5 mm	5 mm
ld	6,4 mm	5 mm	5 mm
gap	0,5 mm	0,5mm	0,7 mm
hmp	2,74 mm	3,44 mm	4,7 mm
Lm	60,2 mm	42,15 mm	32,78 mm

Tableau V.4 – Comparaison des résultats obtenus par les 3 algorithmes.

– Les trois algorithmes obtiennent quasiment les mêmes valeurs pour le facteur de puissance

($\approx 0,8$) et pour rendement ($\approx 0,9$).

- Le couple moteur obtenu par l'algorithme *OMNIOPT* est plus grand que le couple obtenu par les deux autres algorithmes.
- L'algorithme *OMNIOPT* améliore le rapport *RPM* de la solution de référence de 84% tandis que l'algorithme ASREA l'améliore de 21%.

3 Discussion

Dans ce chapitre, nous avons présenté deux modélisations mathématiques du problème d'optimisation d'un moteur électrique à aimant permanent. La première modélisation est une optimisation mono-objectif, tandis que la seconde est une optimisation multiobjectif. Les résultats obtenus en utilisant nos algorithmes d'optimisation sont de très bonne qualité, en comparaison avec les meilleurs résultats obtenus dans ce domaine [Giur 08] (notre gain est de 47% de masse en moins, contre 32% pour [Giur 08], pour les mêmes performances). Les résultats obtenus par la première modélisation restent plus intéressants que pour la seconde modélisation. Nous espérons améliorer dans les prochains mois la modélisation du problème multiobjectif, en affinant le choix des paramètres de chaque algorithme et en ajoutant un troisième objectif, qui aura pour but de minimiser les différentes pertes. Toutes ces améliorations vont nous aider à optimiser deux autres structures du moteur électrique à aimant permanent, qui sont à l'étude actuellement.

Chapitre VI

Métaheuristiques pour l'optimisation continue

Les travaux décrits dans ce chapitre ont donné lieu aux publications suivantes : [Melk 09], [Idou 10f], [Idou 10d], [Idou 11b], [Idou 11a], [Roch 12], [Roch 11a].

Nos deux articles joints à ce mémoire sont :

- **L. Idoumghar**, M. Idrissi-Aouad, M. Melkemi and R. Schott, “Hybrid PSO-SA Type Algorithms for Multi-Modal Function Optimization and Reducing Energy Consumption in Embedded Systems”. Journal of Applied Computational Intelligence and Soft Computing, vol. 2011, pp. 1-12 , juin 2011.
- **L. Idoumghar**, M.Idrissi-Aouad, M. Melkemi and R. Schott, “Metropolis Particle Swarm Optimization Algorithm with Mutation Operator For Global Optimization Problems”. 22th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2010), pp. 35-42, October 27-29, 2010, Arras, France.

Pour mieux calibrer nos algorithmes et étudier leurs comportements, nous avons développé une application, appelée *MetaheuristicDemo*, équipée d'une interface graphique pour visionner les *benchmarks* en 3D ainsi que le processus de recherche suivi par chaque algorithme. Toutes les figures de ce chapitre sont des copies écrans de notre application *MetaheuristicDemo* qui intègre, à ce jour, 33 algorithmes et 75 *benchmarks* mathématiques¹. Cette application est actuellement utilisée par plusieurs collègues pour assurer leurs cours d'optimisation et pour illustrer le comportement de 33 algorithmes sur 75 *benchmarks* mathématiques. Elle est téléchargeable en utilisant le lien suivant :

<http://www.lmia.uha.fr/~mage/idoumghar/MetaheuristicDemo.jar>

L'avantage majeur de cette application est qu'elle permet de calibrer les différents algorithmes lors de la résolution des problèmes en 3D avant d'aborder leur généralisation en *n*-dimensions.

1. Liste des *benchmarks* est définie lors de la conférence CEC'2005 (IEEE)

1 Introduction

Les métaheuristiques, comme nous l'avons mentionné précédemment, ne garantissent pas l'obtention de la meilleure solution. Une discussion peut être alors faite sur le critère de comparaison à définir à propos de diverses métaheuristiques. Depuis 2005, un ensemble de *benchmarks* mathématiques ([[Suga 05](#)], [[Molg 05](#)]) est défini et constitue presque un standard pour la comparaison et l'évaluation des performances des métaheuristiques. Cet ensemble regroupe des fonctions :

- (a) : unimodales, convexes, de dimension élevée,
- (b) : multimodales, à deux dimensions avec un faible nombre d'extrema locaux,
- (c) : multimodales, à deux dimensions avec un nombre élevé d'extrema locaux,
- (d) : multimodale, de dimension élevée, avec un nombre élevé d'extrema locaux.

La catégorie (a) contient des fonctions très pratiques pouvant entraîner un ralentissement de la convergence vers l'extremum global unique. La catégorie (b) est médiane entre (a) et (c)-(d) et est utilisée pour tester la qualité des algorithmes d'optimisation standards dans les environnements hostiles, spécifiquement ceux ayant peu d'extrema locaux avec un extremum global unique. Les catégories (c)-(d) sont considérées comme étant des problèmes de test très difficiles. En effet, les problèmes d'optimisation rencontrés dans la pratique sont souvent de dimensions élevées et sont classés dans la catégorie (d).

À noter que pour la plupart de ces fonctions, l'optimum global est connu. Il est donc possible de compter simplement le nombre d'évaluations ou encore de fixer celui-ci et de discuter sur la meilleure valeur obtenue pour la fonction objectif. Cette dernière démarche est la mieux adaptée aux problèmes réels où l'optimum n'est pas connu à priori.

2 Description de quelques *benchmarks* mathématiques

Dans cette section, nous présentons dix fonctions tests que nous avons utilisés pour comparer nos algorithmes d'optimisation avec quelques algorithmes publiés ces dernières années dans la littérature. La plupart de ces fonctions sont décrites en détail dans [[Suga 05](#)] et [[Molg 05](#)].

Le choix de ces fonctions est fait sur la base de leurs propriétés qui sont similaires aux problèmes du monde réel et fournissent une bonne base pour tester la crédibilité d'un algorithme d'optimisation. Toutes les fonctions tests sont formulées, ci-après, comme étant des problèmes de minimisation. Elles peuvent également être utilisées pour des problèmes de maximisation en inversant simplement le signe de la fonction objectif.

VI.2 Description de quelques *benchmarks* mathématiques

2.1 Fonction de Rastrigin

La fonction de Rastrigin est fortement multimodale et de dimension élevée. Elle est définie mathématiquement par l'équation VI.1 :

$$f_1(\vec{x}) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (\text{VI.1})$$

où \vec{x} est un vecteur de dimension n dont les composantes x_i appartiennent à l'intervalle $[-5.12; 5.12]$. L'emplacement des minima locaux est distribué uniformément. Le minimum global se trouve à l'origine et la valeur de la fonction en ce point est égale à zéro. La figure VI.1 montre une fonction de Rastrigin en deux dimensions.

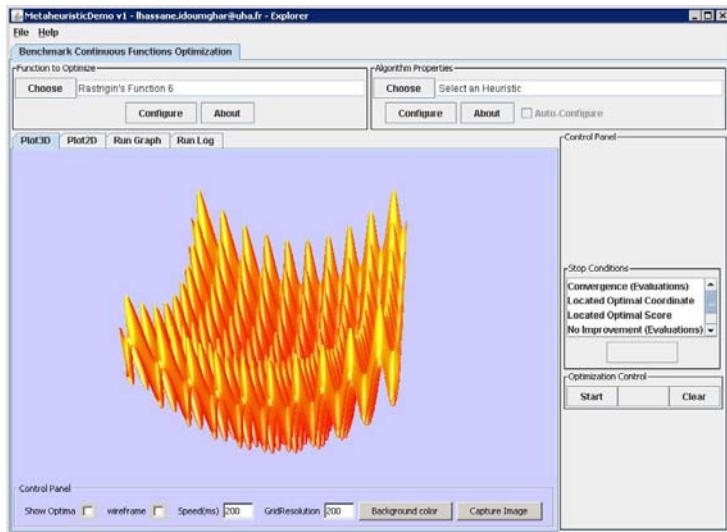


Figure VI.1 – Fonction de Rastrigin en deux dimensions.

2.2 Fonction Sphère

La fonction Sphère est unimodale, convexe, continue et symétrique. Elle est définie par :

$$f_2(\vec{x}) = \sum_{i=1}^n x_i^2 \quad (\text{VI.2})$$

où \vec{x} est un vecteur de dimension n dont les composantes x_i appartiennent à l'intervalle $[-5.12; 5.12]$. Le minimum global se trouve à l'origine et la valeur de la fonction en ce point est égale à zéro. La figure VI.2 montre une fonction Sphère en deux dimensions.

Chapitre VI. Métaheuristiques pour l'optimisation continue

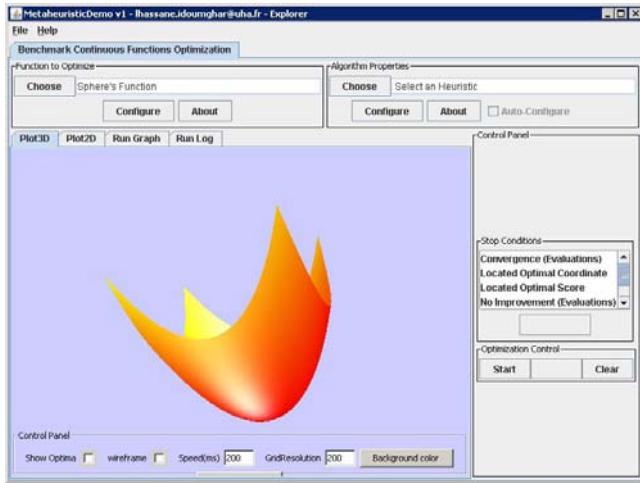


Figure VI.2 – Fonction Sphère en deux dimensions.

2.3 Fonction de Griewank

La fonction de Griewank présente plusieurs minima locaux uniformément distribués sur tout le domaine. Sa formule mathématique est définie par :

$$f_3(\vec{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (\text{VI.3})$$

où \vec{x} est un vecteur de dimension n dont les composantes x_i appartiennent à l'intervalle $[-600; 600]$. Le minimum global se trouve à l'origine et la valeur de la fonction en ce point est égale à zéro. La figure VI.3 montre une fonction de Griewank en deux dimensions.

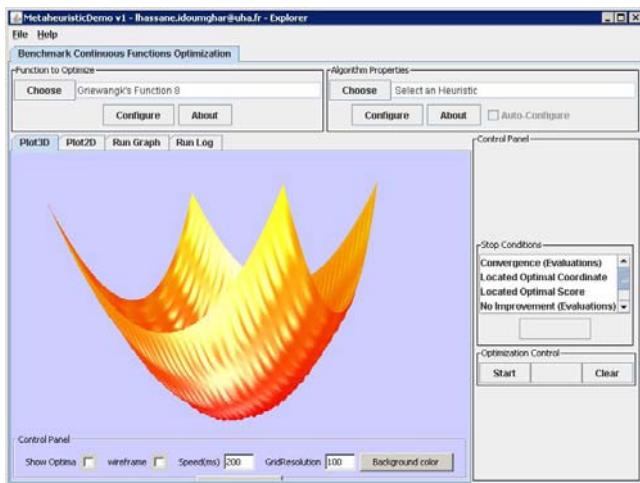


Figure VI.3 – Fonction de Griewank en deux dimensions.

VI.2 Description de quelques *benchmarks* mathématiques

2.4 Fonction de Rosenbrock

La fonction de Rosenbrock est unimodale, mais le minimum se trouve dans une région très étroite, à l'intersection de vallées très peu pentues. Elle est définie par :

$$f_4(\vec{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] \quad (\text{VI.4})$$

où \vec{x} est un vecteur de dimension n dont les composantes x_i appartiennent à l'intervalle $[-30; 30]$. Le minimum global est localisé en $(1, \dots, 1)$ et la valeur de la fonction vaut zéro en ce point. Cette fonction présente une vallée profonde de forme parabolique. Trouver la vallée est trivial, mais parvenir à une convergence vers l'optimum global est une tâche difficile. Dans la littérature, cette fonction est considérée comme étant un problème difficile en raison de l'interaction non linéaire entre les variables [Orti 05]. La figure VI.4 montre une fonction de Rosenbrock en deux dimensions.

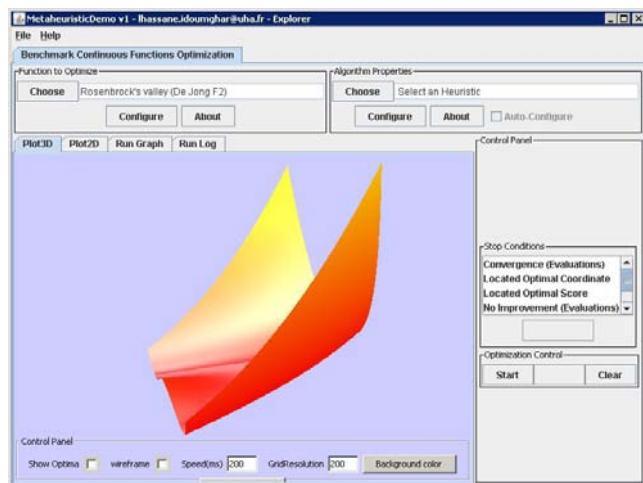


Figure VI.4 – Fonction de Rosenbrock en deux dimensions.

2.5 Fonction bicarrée bruitée

La difficulté de cette fonction est liée au bruit stochastique ajouté. La fonction bicarrée bruitée ou *Noisy* est définie par :

$$f_5(\vec{x}) = \left[\sum_{i=1}^n (i+1)x_i^4 \right] + rand[0, 1] \quad (\text{VI.5})$$

où \vec{x} est un vecteur de dimension n dont les composantes x_i appartiennent à l'intervalle $[-1.28; 1.28]$. La fonction s'annule en ce minimum global.

2.6 Fonction de Schwefel

Cette fonction est également connue sous le nom de *Schwefel's sine root function*. Elle est définie par l'équation VI.6 :

$$f_6(\vec{x}) = - \sum_{i=1}^n \left[x_i \sin(\sqrt{|x_i|}) \right] \quad (\text{VI.6})$$

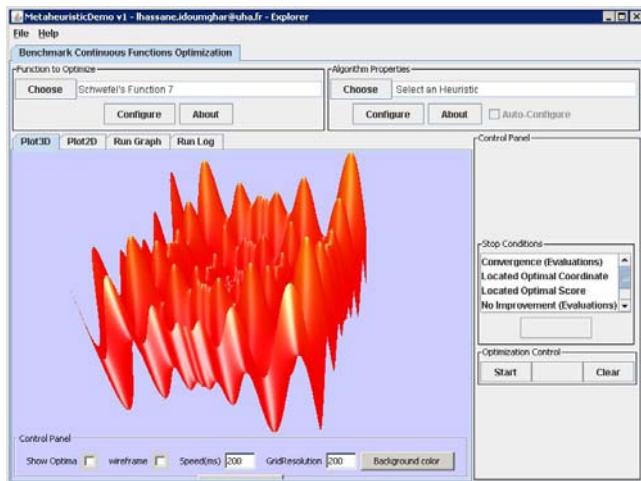


Figure VI.5 – Fonction de Schwefel en deux dimensions.

où \vec{x} est un vecteur de dimension n dont les composantes x_i appartiennent à l'intervalle $[-500; 500]$. La surface de la fonction de Schwefel est composée d'un grand nombre de pics et de vallées. La principale difficulté de cette fonction est que de nombreux algorithmes de recherche sont pris au piège du deuxième meilleur minimum qui est très loin de l'optimum global. En outre, le minimum global est proche des limites du domaine. Les algorithmes de recherche sont potentiellement sujets à converger vers la mauvaise direction dans l'optimisation de cette fonction. La fonction s'annule en ce minimum global (pour f_6 le minimum global est $-420.9687 \times 20 = -8419.368$). La figure VI.5 montre une fonction de Schwefel en deux dimensions.

VI.2 Description de quelques *benchmarks* mathématiques

2.7 Fonction d'Ackley

La fonction d'Ackley, généralisée à n dimensions par [Bddo 96], est définie par :

$$f_7(\vec{x}) = 20 + e - 20 e^{-0.2 (\frac{1}{n} \sum_{i=1}^n x_i^2)^{\frac{1}{2}}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)} \quad (\text{VI.7})$$

où \vec{x} est un vecteur de dimension n dont les composantes x_i appartiennent à l'intervalle $[-32; 32]$. La fonction d'Ackley est hautement multimodale avec des optima locaux uniformément distribués. Le minimum global se trouve à l'origine et la fonction s'annule en ce point. La figure VI.6 montre une fonction d'Ackley en deux dimensions.

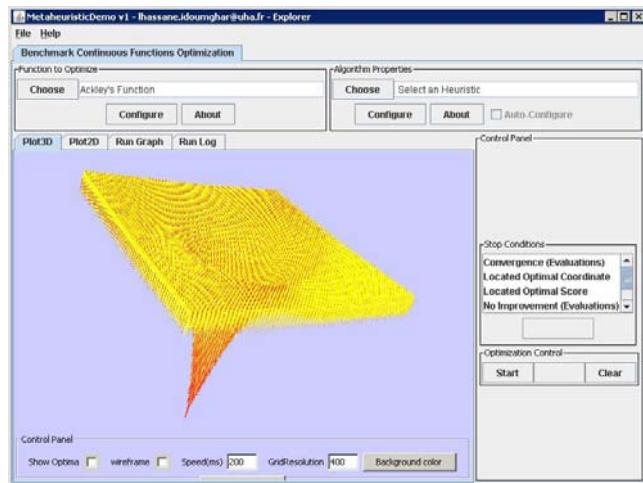


Figure VI.6 – Fonction d'Ackley en deux dimensions.

2.8 Fonction de Michalewicz

La fonction de Michalewicz est paramétrée et multimodale avec plusieurs optima locaux ($n!$) situés entre les plateaux. Le paramètre m définit la “raideur” des vallées ou des bords. Plus le paramètre m est élevé, plus l'optimum global devient difficile à trouver. Dans nos expérimentations, nous prenons $m = 10$. La formule mathématique de la fonction de Michalewicz est donnée par :

$$f_8(\vec{x}) = - \sum_{i=1}^n \sin(x_i) \sin^{2m} \left(\frac{i - x_i^2}{\pi} \right) \quad (\text{VI.8})$$

où \vec{x} est un vecteur de dimension n dont les composantes x_i appartiennent à l'intervalle $[-\pi; \pi]$. La figure VI.7 montre une fonction de Michalewicz en deux dimensions.

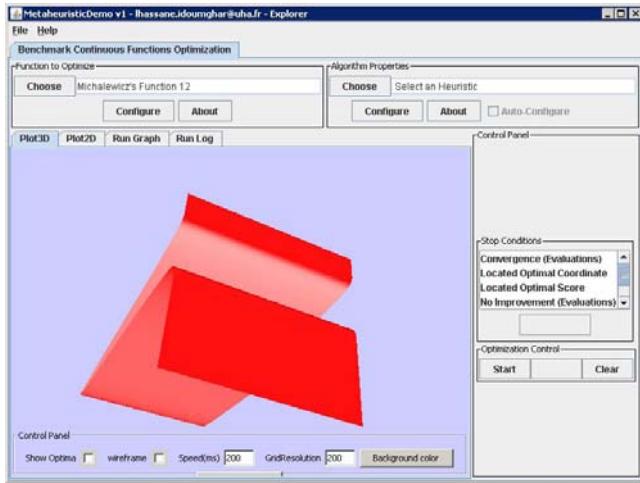


Figure VI.7 – Fonction de Michalewicz en deux dimensions.

2.9 Fonction d'Himmelblau

Dans l'optimisation mathématique, la fonction d'Himmelblau est une fonction multimodale utilisée pour tester les performances des algorithmes d'optimisation. Elle est définie par :

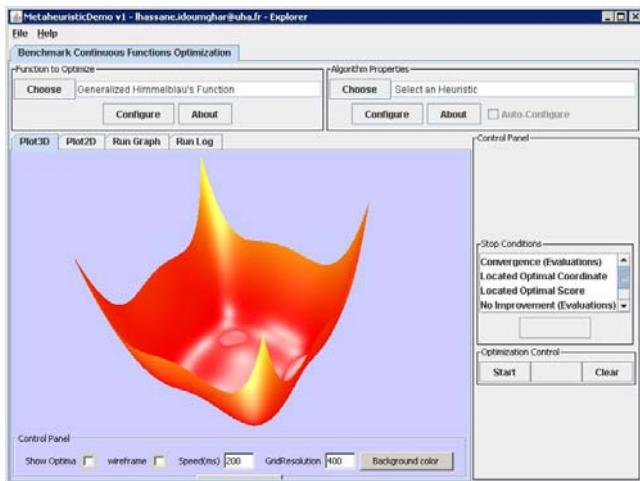


Figure VI.8 – Fonction d'Himmelblau en deux dimensions.

$$f_9(\vec{x}) = (x_2 + x_1^2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + x_1 \quad (\text{VI.9})$$

où $-5 \leq x_i \leq 5$ pour $i = 1, 2$. La valeur de la fonction pour le minimum global vaut -3.78396 . La figure VI.8 montre une fonction d'Himmelblau en deux dimensions.

VI.3 Optimisation des *benchmarks* mathématiques par l'algorithme *PSOSA*

2.10 Fonction de Shubert

C'est une fonction multimodale définie par :

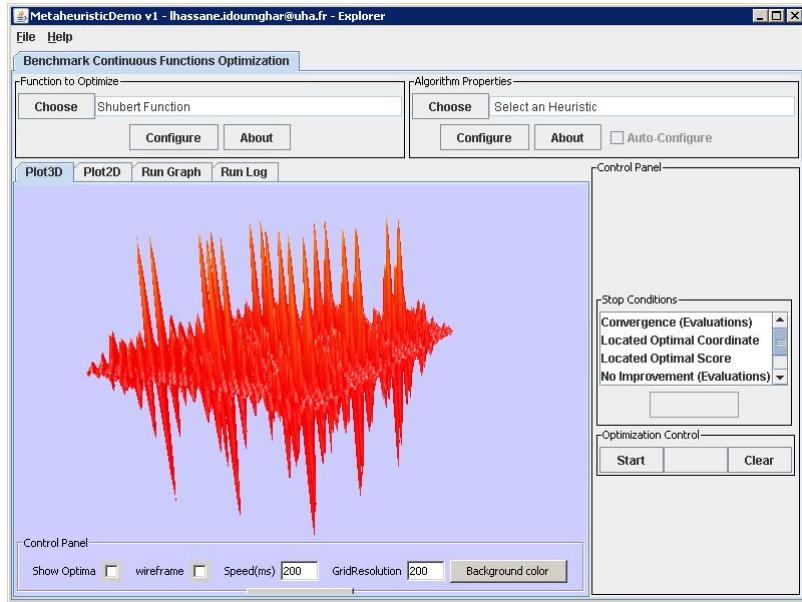


Figure VI.9 – Fonction de Shubert en deux dimensions.

$$f_{10}(\vec{x}) = \sum_{i=1}^5 i \cos((i+1)x_1 + i) \sum_{i=1}^5 i \cos((i+1)x_2 + i) \quad (\text{VI.10})$$

où $-10 \leq x_i \leq 10$ pour $i = 1, 2$. Elle possède 760 minima locaux, dont 18 sont des minima globaux avec une valeur de fonction de -186.7309 . En outre, les optima globaux sont inégalement répartis. La figure VI.9 montre une fonction de Shubert en deux dimensions.

3 Optimisation des *benchmarks* mathématiques par l'algorithme *PSOSA*

Dans cette section nous présentons les résultats de comparaison de notre algorithme *PSOSA* (chapitre I, section 4.2.2, algorithme 11) avec, d'un côté, l'algorithme *TL-PSO* [Naka 08] et avec les algorithmes *PSO* (algorithme standard), *ATREPSO*, *QIPSO* et *GMPSO* [Pant 08] d'un autre côté. Tous ces algorithmes ont été décrits dans la section 2.4.2 du chapitre I.

Nous exécutons notre algorithme hybride sur les fonctions tests mathématiques présentées en section 2. Ces fonctions contiennent beaucoup d'optima locaux dans leur espace de solution.

La quantité d'optima locaux augmente avec la complexité croissante des fonctions, *i.e.* avec l'augmentation de la dimension. Dans nos expérimentations, nous avons utilisé ces fonctions en dimension élevée (20 ou 30 selon l'algorithme considéré), sauf pour le cas des fonctions d'Himmelblau et de Shubert qui sont en deux dimensions par définition.

3.1 Comparaison avec l'algorithme ***TL-PSO***

Comme décrit dans [Naka 08], nous exécutons notre algorithme *PSOSA* sur les quatre fonctions tests mathématiques suivantes : Rastrigin, Schwefel, Griewank et Rosenbrock. Ici, le nombre de particules dans l'essaim est fixé à 30. La dimension de recherche est égale à $n = 20$ et le nombre maximum d'évaluations de la fonction objectif est de 60000 (*i.e.* 2000×30). Les résultats obtenus lors des simulations numériques sont donnés dans le tableau VI.1.

Fonction	Résultats moyens		Meilleurs résultats		Pires résultats	
	TL-PSO	PSOSA	TL-PSO	PSOSA	TL-PSO	PSOSA
Rastrigin	8.7161	5.23034e – 05	4.0589	4.35994e – 09	17.053	8.15835e – 04
Schwefel	445.8830	206.693255	0.6748	0	829.4431	355.318
Griewank	0.0228	1.18294e – 03	1.0507e – 5	1.2299e – 15	0.0739	0.0172263
Rosenbrock	19.5580	0.58359742	0.1331	0.0442625	71.5439	1.36171

Tableau VI.1 – Comparaison entre *PSOSA* et *TL-PSO*.

Ces résultats indiquent les valeurs moyennes, les meilleures valeurs ainsi que les pires valeurs obtenues sous la même condition de 50 essais. En analysant le tableau VI.1, nous pouvons constater que les résultats obtenus avec notre algorithme *PSOSA* sont excellents en comparaison avec ceux qui sont obtenus par l'algorithme *TL-PSO*.

3.2 Comparaison avec les algorithmes ***PSO*, *ATREPSO*, *QIPSO* et *GMPSO***

Pour effectuer une comparaison équitable entre *PSO*, *ATREPSO*, *QIPSO*, *GMPSO* et notre algorithme *PSOSA*, nous avons fixé, comme indiqué dans [Pant 08], la même méthode pour la génération aléatoire de nombres de telle sorte que la population initiale de l'essaim soit la même pour tous les cinq algorithmes. Le nombre de particules de l'essaim est fixé à 30. Les cinq algorithmes utilisent un poids d'inertie ω , linéairement décroissant, commençant à 0.9 et se terminant à 0.4, avec des paramètres définis par l'utilisateur $c_1 = c_2 = 2.0$. Pour chaque

VI.3 Optimisation des *benchmarks* mathématiques par l'algorithme *PSOSA*

algorithme, le nombre maximum d'évaluations de la fonction objectif est de 300000. Pour chaque *benchmark* mathématique étudié, 30 exécutions ont été effectuées et la moyenne des meilleures solutions trouvées a été enregistrée. La solution moyenne et l'écart-type² trouvés par les cinq algorithmes sont donnés dans le tableau VI.2.

Fonction	PSO	QIPSO	ATREPSO	GMPSO	PSOSA
Rastrigin	22.339158	11.946888	19.425979	20.079185	0
	15.932042	9.161526	14.349046	13.700202	0
Sphère	$1.167749e - 45$	0.000000	$4.000289e - 17$	$7.263579e - 17$	$5.3656e - 32$
	$5.222331e - 46$	0.000000	0.000246	$6.188854e - 17$	$2.98492e - 31$
Griewank	0.031646	0.011580	0.025158	0.024462	$3.32255e - 20$
	0.025322	0.012850	0.028140	0.039304	$2.68415e - 20$
Rosenbrock	22.191725	8.939011	19.490820	14.159547	0.227048188
	$1.615544e + 04$	3.106359	$3.964335e + 04$	$4.335439e + 04$	0.243978057
Noisy	8.681602	0.451109	8.046617	7.160675	0.002019998
	9.001534	0.328623	8.862385	7.665802	0.000650347
Schwefel	-6178.559896	-6355.586640	-6183.677600	-6047.670898	-8379.66
	$4.893329e + 02$	477.532584	469.611104	482.926738	$2.20425e - 19$
Ackley	$3.483903e - 18$	$2.461811e - 24$	0.018493	$1.474933e - 18$	$7.43546e - 16$
	$8.359535e - 19$	0.014425	0.014747	$1.153709e - 08$	$1.09382e - 15$
Michalewicz	-18.159400	-18.469600	-18.982900	-18.399800	-19.62555806
	1.051050	0.092966	0.272579	0.403722	0.00926944
Himmelblau	-3.331488	-3.783961	-3.751458	-3.460233	-3.78396
	1.243290	0.190394	0.174460	0.457820	$3.16001e - 15$
Shubert	-186.730941	-186.730942	-186.730941	-186.730942	-186.730942
	$1.424154e - 05$	0.000000	$1.424154e - 05$	$1.525879e - 05$	$8.66746e - 14$

Tableau VI.2 – Comparaison des moyennes/écart-types des solutions obtenues en utilisant *PSOSA*, *PSO*, *QIPSO*, *ATREPSO* et *GMPSO*.

Les résultats numériques donnés dans le tableau VI.2 montrent que :

- Tous les algorithmes dépassent l'algorithme *PSO* classique.
- *PSOSA* a de meilleures performances en comparaison avec *PSO*, *QIPSO*, *ATREPSO* et *GMPSO*, excepté pour les fonctions Sphère et Ackley.
- Pour la fonction Sphère, *QIPSO* obtient de meilleurs résultats que ceux qui sont obtenus par *PSOSA*, mais, lorsque le nombre maximal d'itérations est fixé à 1.5×10^6 , *PSOSA* trouve la valeur optimale.
- Pour la fonction d'Ackley, *QIPSO* obtient de meilleurs résultats que *PSOSA*. Mais, *PSOSA* a un écart-type plus petit que celui de *QIPSO*.

2. Notons que l'écart-type donne une indication sur la stabilité des algorithmes.

3.3 Tests statistiques

Les tests statistiques constituent des outils d'aide à la décision. Ils permettent de trancher entre deux hypothèses dont une et une seule est vraie. Avant d'aborder le test de Friedman que nous avons utilisé pour réaliser nos tests, nous avons jugé intéressant de définir certains concepts utiles à la compréhension des tests statistiques.

Définition 16

Une *hypothèse statistique* est un énoncé concernant les caractéristiques (valeurs des paramètres, forme de la distribution des observations) d'une population.

Définition 17

Hypothèses statistiques : l'hypothèse selon laquelle on fixe a priori un paramètre de la population à une valeur particulière s'appelle l'*hypothèse nulle*, notée H_0 . Toute autre hypothèse qui diffère de l'hypothèse H_0 s'appelle l'*hypothèse alternative* et est notée H_1 .

Définition 18

Un *test d'hypothèse* est une méthode qui a pour but de fournir une règle de décision permettant, sur la base de résultats d'échantillon, de faire un choix entre deux hypothèses statistiques.

Un des objectifs d'un test d'hypothèse est de prédire d'avance à quelle condition l'une ou l'autre des hypothèses sera considérée comme *vraisemblable*. C'est H_0 qui est soumise au test et toute la démarche du test s'effectue en considérant cette hypothèse comme vraie. Si le test conduit au rejet de H_0 , nous considérons alors H_1 comme vraisemblable plutôt que H_0 .

Pour établir la crédibilité de H_0 , il faut être en mesure d'établir des règles de décision qui vont aboutir sans équivoque au non-rejet de H_0 (ou au rejet). En pratique, on met en œuvre une démarche qui permet, à long terme, de rejeter à tort H_0 vraie dans une faible proportion de cas. La conclusion qui est déduite des résultats de l'échantillon suivant la règle de décision adoptée, a un caractère probabiliste. On ne prend une décision qu'en prenant conscience qu'il y a un certain risque qu'elle soit erronée. Ce risque est donné par le *seuil de signification* du test.

Définition 19

Le seuil de signification d'un test d'hypothèse est le risque consenti à l'avance de rejeter à tort H_0 alors qu'elle est vraie (et de favoriser alors H_1). Ce seuil de signification s'exprime en probabilité comme suit :

$$\alpha = \text{Probabilité [rejeter } H_0 \mid H_0 \text{ vraie}] = \text{Probabilité [choisir } H_0 \mid H_1 \text{ vraie}] \quad (\text{VI.11})$$

VI.3 Optimisation des *benchmarks* mathématiques par l'algorithme *PSOSA*

À ce seuil de signification, on peut faire correspondre, sur la distribution d'échantillonnage de la statistique, une région de rejet de H_0 (appelée également *région critique*). L'aire de cette région correspond à la probabilité α (souvent fixée à 5% (parfois à 1%)). Cette région de rejet de H_0 est constituée d'un ensemble de valeurs de la statistique qui conduiront au rejet de H_0 . Par exemple, si on prend comme seuil de signification $\alpha = 5\%$, cela signifie que l'on admet d'avance que la statistique peut prendre, dans 5% des cas, une valeur se situant dans la région de rejet de H_0 , bien que l'hypothèse H_0 soit vraie et ceci uniquement d'après le hasard de l'échantillonnage. À la distribution d'échantillonnage correspond aussi une région complémentaire, dite région de non-rejet de H_0 (appelée également *région d'acceptation*) de probabilité $1 - \alpha$. La valeur observée de la statistique déduite des résultats de l'échantillon appartient, soit à la région de rejet de H_0 , soit à la région de non-rejet de H_0 .

3.3.1 Test de Friedman

Afin de comparer les performances des algorithmes *PSO*, *ATREPSO*, *QIPSO*, *GMPSO* et notre algorithme *PSOSA*, nous avons effectué le test de Friedman ([Vill 11], [Garc 09]).

Avant de commencer notre analyse des résultats, rappelons les deux hypothèses statistiques de ce test :

- H_0 : *il n'y aucune différence entre les métaheuristiques étudiées.*
- H_1 : *au moins une métaheuristique est différente des autres.*

Pour effectuer le test de Friedman, nous commençons par déterminer le rang de chaque métaheuristique pour chaque *benchmark* mathématique étudié. Le tableau VI.3 donne les rangs des algorithmes en fonction des moyennes obtenues sur chaque *benchmark* mathématique.

A partir des valeurs stockées dans le tableau VI.3, on calcule A_2 et B_2 . A_2 est la somme totale des carrés des rangs :

$$A_2 = \sum_{i=1}^b \sum_{j=1}^k [R(X_{ij})]^2$$

$$A_2 = 119.25 + 36 + 140.25 + 133 + 39 = 547.5$$

et B_2 est donné par la relation suivante :

$$B_2 = \frac{1}{b} \sum_{j=1}^k R_j^2$$

$$B_2 = \frac{1}{10} (43.5^2 + 18^2 + 36.5^2 + 35^2 + 17^2) = 506.25$$

Instance (b)	PSO			QIPSO			ATREPSO		
	X_{b1}	$R(X_{b1})$	$R(X_{b1})^2$	X_{b2}	$R(X_{b2})$	$R(X_{b2})^2$	X_{b3}	$R(X_{b3})$	$R(X_{b3})^2$
f_1	22.339158	5	25	11.946888	2	4	19.425979	3	9
f_2	1.167749e-45	2	4	0.000000	1	1	4.000289e-17	4	16
f_3	0.031646	5	25	0.011580	2	4	0.025158	4	16
f_4	22.191725	5	25	8.939011	2	4	19.490820	4	16
f_5	8.681602	5	25	0.451109	2	4	8.046617	4	16
f_6	-6178.559896	4	16	-6355.586640	2	4	-6183.677600	3	9
f_7	3.483903e-18	3	9	2.461811e-24	1	1	0.018493	5	25
f_8	-18.159400	5	25	-18.469600	3	9	-18.982900	2	4
f_9	-3.331488	5	25	-3.783961	1	1	-3.751458	3	9
f_{10}	-186.730941	4,5	20,25	-186.730942	2	4	-186.730941	4,5	20,25
Moyenne des rangs	4,35			1,8			3,65		
Somme totale	43,5			18			36,5		

Instance (b)	GMPSO			PSOSA		
	X_{b4}	$R(X_{b4})$	$R(X_{b4})^2$	X_{b5}	$R(X_{b5})$	$R(X_{b5})^2$
f_1	20.079185	4	16	0,0000E+00	1	1
f_2	7.263579e-17	5	25	5.3656e-32	3	9
f_3	0.024462	3	9	3.32255e-20	1	1
f_4	14.159547	3	9	0.227048188	1	1
f_5	7.160675	3	9	0.002019998	1	1
f_6	-6047.670898	5	25	-8379.66	1	1
f_7	1.474933e-18	2	4	7.43546e-16	4	16
f_8	-18.399800	4	16	-19.62555806	1	1
f_9	-3.460233	4	16	-3.78396	2	4
f_{10}	-186.730942	2	4	-186.730942	2	4
Moyenne des rangs	3,5			1,7		
Somme totale	35			133		

Tableau VI.3 – Classement des métaléuristiques

	QIPSO	ATREPSO	GMPSO	PSOSA
PSO	25,5	7	8,5	26,5
QIPSO	-	18,5	17	1
ATREPSO	-	-	1,5	19,5
GMPSO	-	-	-	18

Tableau VI.4 – Test bilatéral

VI.3 Optimisation des *benchmarks* mathématiques par l'algorithme *PSOSA*

où b est le nombre de *benchmark* mathématiques étudiés et k le nombre de métaheuristiques à comparer.

Ensuite, on calcule le test statistique T_2 en utilisant la formule suivante :

$$T_2 = \frac{(b-1)[B_2 - bk(k+1)^2/4]}{A_2 - B_2}$$

$$T_2 = 12.27$$

En utilisant la table de la distribution F , avec le $\alpha = 5\%$, on obtient la valeur suivante :

$$F_{1-\alpha, k-1, (b-1)(k-1)} = F_{0.95, 4, 36} = 3.39$$

Étant donné que $T_2 > F_{0.95, 4, 36}$, l'hypothèse H_0 est rejetée, *i.e.* il existe au moins une métaheuristique dont les performances sont différentes d'au moins une des autres métaheuristiques étudiées.

Pour déterminer quelle métaheuristique est différente des autres, nous avons réalisé le test bilatéral et les résultats sont donnés dans le tableau VI.4. En effet, deux algorithmes A et B sont différents si $|R_A - R_B|$ est supérieur à la valeur critique C , où R_i est la somme des rangs de l'algorithme i . La valeur critique C est calculée en utilisant l'expression suivante :

$$C = t_{1-\frac{\alpha}{2}} \left[\frac{2b(A_2 - B_2)}{(b-1)(k-1)} \right]^{\frac{1}{2}}$$

où $t_{1-\frac{\alpha}{2}}$ est le $(1 - \frac{\alpha}{2})^{\text{ème}}$ coefficient de la distribution t et $(b-1)(k-1)$ correspond au nombre de degrés de liberté du système.

Dans notre cas, la valeur critique $C = 2.71 \left[\frac{2 \times 5 \times (547.5 - 506.25)}{9 \times 4} \right]^{\frac{1}{2}} = 13.01$.

Le tableau VI.4 montre que les résultats obtenus par l'algorithme *PSOSA* sont différents de ceux obtenus par les algorithmes *PSO*, *ATREPSO* et *GMPSO*, tandis qu'il obtient les mêmes performances que celles de l'algorithme *QIPSO*. A partir de cette analyse, nous pouvons dire que *PSOSA* obtient de meilleurs résultats par rapport aux métaheuristiques suivantes : *PSO*, *ATREPSO* et *GMPSO*.

4 Optimisation des *benchmarks* mathématiques par l'algorithme *MPSOM*

4.1 Comparaison avec les algorithmes *GPSO-J*, *CLPSO* et *ATM-PSO*

Nous avons comparé notre algorithme *MPSOM* (décrit dans le chapitre III, section 3.1.1) avec les algorithmes suivants : l'algorithme *Gaussian PSO with jumps* (*GPSO-J*) [Kroh 05], l'algorithme *Comprehensive Learning PSO* (*CLPSO*) [Lian 06] et l'algorithme *Particle Swarm Optimizer with Adaptive Tabu and Mutation* (*ATM-PSO*) [Yu X 10]. Tous ces algorithmes ont été décrits en section 2.4.2 du chapitre I. Nous avons exécuté *MPSOM* sur sept fonctions tests mathématiques parmi celles présentées en section 2. Le tableau VI.5 récapitule leurs caractéristiques.

Tableau VI.5 – Synthèse des fonctions tests mathématiques utilisées.

Fonction	Problème	Rang	$f(x^*)$	ϵ	Classification
Sphère	$\sum_{i=1}^n x_i^2$	[−100; 100]	0	0.01	Unimodale
Rastrigin	$\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[−5.12; 5.12]	0	10	Multimodale
Griewank	$\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	[−600; 600]	0	0.1	Multimodale
Rosenbrock	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[−2.048; 2.048]	0	100	Unimodale
Schwefel	$420.9687 n - \sum_{i=1}^n \left[x_i \sin(\sqrt{ x_i }) \right]$	[−500; 500]	0	2000	Multimodale
Ackley	$20 + e - 20 e^{-0.2 (\frac{1}{n} \sum_{i=1}^n x_i^2)^{\frac{1}{2}}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)}$	[−30; 30]	0	0.1	Multimodale
Noisy	$\left[\sum_{i=1}^n (i+1)x_i^4 \right] + rand[0, 1]$	[−1.28; 1.28]	0	0.1	Bruitée

Nous avons fixé, comme c'est indiqué dans [Yu X 10], le nombre de particules dans l'essaim à 20 et le nombre maximal d'évaluations de la fonction objectif à 5000 $D = 150000$. Pour chaque *benchmark* mathématique, 30 exécutions ont été effectuées et la moyenne des meilleures solutions trouvées a été enregistrée. Le tableau VI.6 donne les résultats obtenus.

L'analyse des résultats numériques donnés dans le tableau VI.6 montre que :

VI.4 Optimisation des *benchmarks* mathématiques par l'algorithme *MPSOM*

Fonction	CLPSO	GPSO-J	ATM-PSO	MPSOM
Sphère	$8.99e - 14$	$3.85e - 07$	$8.90e - 104$	$9.4008e - 113$
	$\pm 4.66e-14$	$\pm 7.97e-07$	$\pm 3.18e-103$	$\pm 9.6549e-113$
Rosenbrock	20.88	27.11	15.13	0.014441893
	± 2.58	± 16.91	$\pm 8.71e-01$	± 0.016312307
Schwefel	$1.76e - 12$	1336.29	$2.36e - 12$	4501.7
	$\pm 3.27e-13$	± 290.81	$\pm 9.57e-13$	± 1520.45862
Rastrigin	$1.34e - 06$	15.86	0	0
	$\pm 1.66e-06$	± 4.52	± 0	± 0
Ackley	$8.45e - 08$	$1.43e - 03$	$2.59e - 14$	$2.60232e - 10$
	$\pm 1.96e-08$	$\pm 1.36e-03$	$\pm 6.12e-15$	$\pm 5.27812e-11$
Griewank	$1.95e - 09$	$4.02e - 02$	$2.22e - 02$	0
	$\pm 4.35e-09$	$\pm 4.02e-02$	$\pm 2.03e-02$	± 0
Noisy	$8.18e - 03$	$4.45e - 03$	$9.77e - 03$	$4.0721e - 06$
	$\pm 2.39e-03$	$\pm 9.95e-04$	$\pm 3.16e-03$	$\pm 1.055e-05$

Tableau VI.6 – Résultats moyens et écart-types obtenus sur 30 exécutions indépendantes des algorithmes *CLPSO*, *GPSO-J*, *ATM-PSO* et *MPSOM* sur sept fonctions tests mathématiques (dimension $n = 30$ et taille de l'essaim *swarm_size* = 20).

- Tous les algorithmes résolvent avec succès la fonction Sphère qui est la fonction unimodale la plus simple même si notre algorithme *MPSOM* obtient les meilleurs résultats.
- *MPSOM* dépasse les autres algorithmes quand il s'agit d'optimiser les problèmes multimodaux difficiles.
- *ATM-PSO* et *CLPSO* réussissent avec succès à résoudre la fonction de Schwefel. Par contre, les algorithmes *GPSO-J* et *MPSOM* échouent dans l'optimisation de cette fonction.
- Pour la fonction hautement multimodale de Rastrigin, *ATM-PSO* et *MPSOM* atteignent la solution optimale globale après 150000 évaluations de la fonction objectif.
- L'analyse des résultats obtenus pour la fonction d'Ackley montre que *ATM-PSO* et *MPSOM* obtiennent de meilleurs résultats moyens que *GPSO-J* et *CLPSO*.
- L'optimisation de la fonction de Griewank est un exemple d'échec de tous les algorithmes (qui restent bloquer dans des optima locaux), excepté notre algorithme *MPSOM* qui atteint la solution optimale globale.
- L'optimisation de la fonction bicarrée bruitée (*Noisy*) est un autre exemple du succès de *MPSOM* et illustre que notre algorithme est adapté à un environnement dynamique.

4.2 Taux de convergence $Q_{measure}$

Pour évaluer le taux de convergence de tous les algorithmes comparés, un seuil ϵ est positionné pour chaque fonction test comme cela est décrit dans [Yu X 10]. Lorsque chaque algorithme atteint le seuil ϵ spécifié (voir le tableau VI.5) pour une certaine fonction test dans le k^{ieme} essai, le nombre d'évaluations de fonction FE_k nécessaire est enregistré et l'essai courant k est noté comme étant un essai réussi. [Feok 06] propose un critère noté $Q_{measure}$, qui incorpore la mesure de la convergence et de la robustesse. Le $Q_{measure}$ utilisé afin d'évaluer les performances des algorithmes, est défini par l'équation VI.12 où n_t est le nombre total d'essais, n_s le nombre total d'essais réussis et le ratio de succès SR est défini avec $SR = \frac{n_s}{n_t}$.

$$Q_{measure} = \frac{n_t \sum_{i=1}^{n_s} FE_i}{n_s^2} \quad (\text{VI.12})$$

Fonction	CLPSO	GPSO-J	ATM-PSO	MPSOM
Sphère	67977	19343	8101	216.66
	(100%)	(100%)	(100%)	(100%)
Rosenbrock	34654	6537	980	700.66
	(100%)	(100%)	(100%)	(100%)
Schwefel	24439	47987	9726	21271
	(100%)	(100%)	(100%)	(100%)
Rastrigin	87201	1795500	15187	3248
	(100%)	(7%)	(100%)	(100%)
Ackley	62659	42857	9664	4832
	(100%)	(100%)	(100%)	(100%)
Griewank	65437	18309	7140	5085
	(100%)	(90%)	(100%)	(100%)
Noisy	38094	11137	5988	1177.33
	(100%)	(100%)	(100%)	(100%)

Tableau VI.7 – Valeurs de $Q_{measure}$ des algorithmes CLPSO, GPSO-J, ATM-PSO et MPSOM sur sept fonctions tests mathématiques. Les pourcentages en parenthèses sont les ratios de succès (SR).

Les valeurs de $Q_{measure}$ des quatre algorithmes étudiés sont données pour les sept fonctions tests mathématiques dans le tableau VI.7. L'analyse des résultats de ce tableau montre que MPSOM converge plus rapidement que tous les autres algorithmes, excepté pour l'optimisation de la fonction de Schwefel où ATM-PSO fait mieux que MPSOM. La figure VI.10 illustre une trace d'exécution de l'algorithme MPSOM pour la fonction Rastrigin en deux et vingt dimensions.

VI.4 Optimisation des *benchmarks* mathématiques par l'algorithme MPSOM

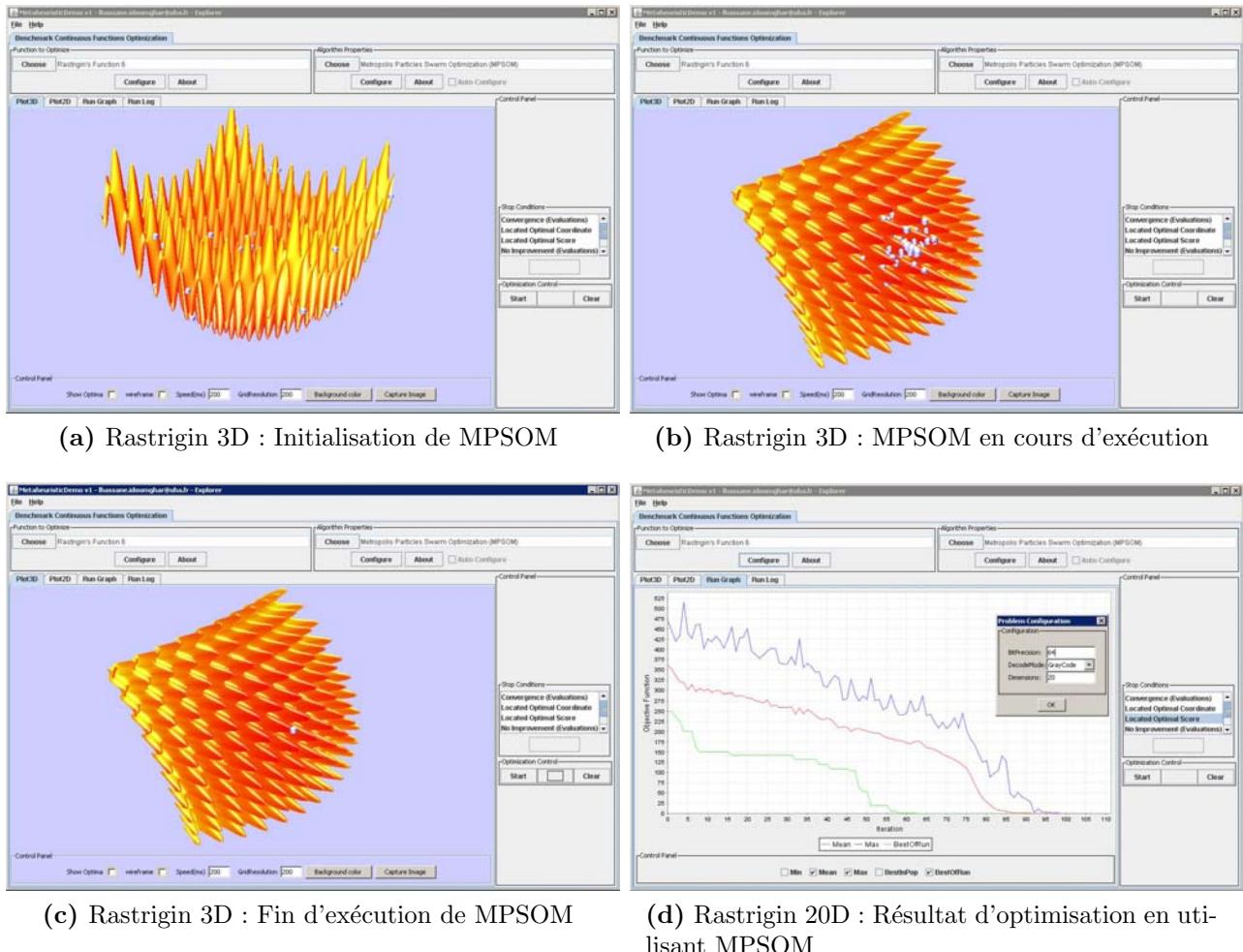


Figure VI.10 – Résultats de l'optimisation de la fonction Rastrigin en 3D et 20D.

4.3 Test de Friedman

Nous avons effectué le test de Friedman sur ces *benchmarks* mathématiques. Le tableau VI.8 donne le rang de chaque métaheuristique selon les moyennes obtenues pour chaque fonction test.

Pour ces *benchmarks* mathématiques, nous obtenons une valeur de $A_2 = 209.5$, $B_2 = 192.92$, $T_2 = 6.49$ et $F_{0.95,2,18} = 3.159$. Avec $T_2 > F_{0.95,2,18}$, l'hypothèse nulle H_0 est rejetée, donc il y a au-moins un algorithme dont la performance est différente d'au moins un des autres algorithmes. Pour savoir quels algorithmes sont différents nous effectuons des tests bilatéraux. La valeur critique est de $C = 7.5$.

Instance (b)	CLPSO			GPSO-J			ATM-PSO			MPSOM		
	X_{b1}	$R(X_{b1})$	$R(X_{b1})^2$	X_{b2}	$R(X_{b2})$	$R(X_{b2})^2$	X_{b3}	$R(X_{b3})$	$R(X_{b3})^2$	X_{b4}	$R(X_{b4})$	$R(X_{b4})^2$
f_1	20.88	3	9	27.11	4	16	15.13	2	4	0.0144418	1	1
f_2	8.99E-14	3	9	3.85E-07	4	16	8.90E-104	2	4	9.40E-113	1	1
f_3	1.76E-12	1	1	1336.29	4	16	2.36E-12	2	4	4501.7	3	9
f_4	1.34E-06	3	9	15.86	4	16	1.00E-46	1.5	2.25	1.00E-46	1.5	2.25
f_5	8.18E-03	3	9	4.45E-03	2	4	9.77E-03	4	16	4.07E-06	1	1
f_6	8.45E-08	3	9	1.43E-03	4	16	2.59E-14	1	1	2.60E-10	2	4
f_7	1.95E-09	2	4	4.02E-02	4	16	2.22E-02	3	9	1.00E-46	1	1
Moyenne		2.57		3.71		26	100		2.21		1.5	
Somme		18	50					15.5	40.25		10.5	19.25

Tableau VI.8 – Classement des métaheuristiques

	GPSO-J	ATM-PSO	MPSOM
CLPSO	8	2.5	7.5
GPSO-J	-	10.5	15.5
ATM-PSO	-	-	5.0

Tableau VI.9 – Test bilatéral

Les tests bilatéraux (voir le tableau VI.9) montrent que les résultats obtenus par l'algorithme *MPSOM* sont différents de ceux obtenus par les algorithmes *CLPSO* et *GPSO – J*. Tandis qu'il obtient les mêmes performances que celles de l'algorithme *ATM – PSO*. A partir de cette analyse, nous pouvons dire que *MPSOM* obtient de meilleurs résultats par rapport aux métaheuristiques suivantes : *CLPSO* et *GPSO – J*.

5 Conclusion

Dans ce chapitre, nous avons présenté les résultats de comparaison de nos algorithme *PSOSA* et *MPSOM* avec quelques algorithmes de la littérature. L'analyse statistique montre que nos algorithmes obtiennent de bons résultats en comparaison avec les algorithmes de la littérature. Dans les perspectives de ce travail, nous envisageons étendre nos tests à d'autres benchmarks mathématiques et de proposer des versions *GPU* de nos algorithmes.

Chapitre VI. Métaheuristiques pour l'optimisation continue

Conclusion et projet scientifique

Dans ce mémoire, j'ai présenté mes principales contributions à la résolution informatique de certains problèmes industriels et académiques, tout en identifiant certaines questions ouvertes et pistes de recherche qui me semblent intéressantes à étudier. En soi, chaque chapitre ouvre des perspectives de travail relativement nombreuses, tant du point de vue pratique que théorique. En effet, le développement des météahuristiques pour la résolution de problèmes d'optimisation constitue mon principal axe de recherche au sein de l'équipe MAGE. Dans cet axe, je m'attache à approfondir davantage les coopérations transversales que j'ai pu mettre en place sur le thème de l'optimisation. Ce qui me permettra de proposer des algorithmes de résolution plus génériques et tenant compte des caractéristiques communes aux problèmes rencontrés, qui sont le plus souvent issus de cas réels. Ces problèmes, généralement NP-difficiles, requièrent l'utilisation de météahuristiques robustes, aisément configurables et pouvant être réutilisées ou combinées suivant les cas concrets traités.

Comme projet scientifique, je propose de poursuivre et d'approfondir ma démarche initiale prenant en compte à la fois des besoins en termes de méthode de conception et en termes d'efficacité des approches développées. Ainsi, mes futurs travaux s'orienteront vers l'utilisation combinée systématique de météahuristiques, principalement pour la prise en compte simultanée de plusieurs objectifs à optimiser ou plusieurs contraintes multiples à satisfaire. Généralement, les méthodes de recherche locale sont manipulées en tant qu'opérateurs dans une approche à base de population (essaim particulaire, algorithme évolutionnaires). Je compte systématiser cette pratique, tout en y incorporant de nouveaux concepts issus du domaine de l'intelligence artificielle et pas encore suffisamment exploités à mon sens en optimisation combinatoire. La combinaison des algorithmes évolutionnaires et des algorithmes de recherche locale pour l'allocation de fréquences en radiodiffusion en est un exemple que nous avons déjà mis en œuvre. L'introduction des techniques d'apprentissage (par renforcement ou par des réseaux de neurones par exemple), pour la recherche locale adaptative est à envisager. De même, la formulation de

schémas heuristiques dans un cadre plus large, tel que celui des systèmes multi-agents, peut présenter des avantages, notamment celui d'offrir un support conceptuel unifié de description tout en favorisant l'implantation parallèle des solutions réalisées.

Dans la suite de ce chapitre, je voudrais décrire quelques pistes plus précises dans lesquelles je souhaite continuer mon travail.

– Méthodes d'optimisation multiobjectif pour la planification des réseaux de radiodiffusion

Dans le chapitre II, nous avons présenté une nouvelle modélisation du problème d'allocation de fréquences dans le domaine de la radiodiffusion. Cette modélisation utilise le calcul par histogramme et une méthode tabou probabilisée pour obtenir des plans de fréquences de très bonne qualité. Actuellement, nous travaillons sur une nouvelle modélisation du problème d'allocation de fréquences pour prendre en compte les futures évolutions du réseau de radiodiffusion. En effet, jusqu'ici, la planification permettait de gérer un type de réseau, par bande en allouant des fréquences à chacun des sites, sans notion du service diffusé. La densification des réseaux et l'évolution des besoins nécessitent de revoir ce mode de fonctionnement de façon à permettre une allocation de fréquences multiservices, prenant éventuellement en compte des notions de niveau de qualité de service distincts. La question de la gestion des réseaux de radiodiffusion mono-fréquences dits SFN sera elle aussi abordée, du fait de la problématique que représentent ces réseaux. Une modélisation spécifique devra être mise en place afin d'appréhender la gestion des retards intra SFN, dès l'allocation. Un agent permettra par ailleurs d'identifier, dès le début du processus de planification, les situations difficiles à traiter. Un sujet de thèse CIFRE vient de démarrer sur ce problème et vise à explorer des aspects qui seront cruciaux pour les futurs réseaux de radiodiffusion ou radiocommunication à horizon de quatre à cinq ans. Le plan France Numérique prévoit d'ailleurs une densification du spectre UHF à hauteur de onze réseaux de télévision fixes et deux réseaux de télévision mobiles. Conjugué à un besoin de révision de l'utilisation du spectre (émergence possible d'une seconde dividende numérique), ces ambitions nécessitent aujourd'hui de renforcer l'efficacité des outils de planification existants sur la base de solutions innovantes. Dans ce travail de thèse, de M. Akram Bedoui qui a démarré en janvier 2012, on s'intéressera entre autres à l'étude et à la satisfaction des contraintes générées par la cohabitation de plusieurs réseaux dans la même bande de fréquences ou dans des bandes adjacentes. Ces nouvelles contraintes de cohabitation seront ajoutées aux contraintes propres à chaque réseau étudié. Le problème ainsi posé pourra être vu comme un problème d'optimisation multiobjectif, en respectant les contraintes

de chaque réseau. Il est à noter que l'allocation automatique de fréquences passe par une étape préalable impliquant des calculs d'histogrammes (nombre de points couverts pour chaque écart de fréquences) sur des grandes zones, afin de mieux estimer la zone de couverture pour chaque émetteur. Il est donc primordial de disposer de modules efficaces pour ces calculs. Le calcul de propagation et l'allocation de fréquences sont réellement deux points clés indissociables du processus de planification d'un réseau radioélectrique. La solution idéale est d'intégrer le module de propagation dans le processus de l'allocation de fréquences. Se pose alors le problème du temps de calcul qui croît rapidement, d'où la nécessité d'avoir recours à des algorithmes massivement parallèles. A l'issue de ce travail, nous espérons proposer des techniques innovantes d'optimisation multiobjectif et les tester sur des réseaux de radiodiffusion et radiocommunication intéressant particulièrement TDF, notre partenaire industriel. Le cas de l'allocation statique sera abordé en premier lieu puis, les techniques d'allocation dynamique seront étudiées.

– Planification de la production de l'énergie dans les *smart grids*

Nous avons proposé dans le chapitre III une nouvelle architecture pour la gestion intelligente de l'énergie dans les *smart grids*. Une première évaluation des besoins, en termes de nombre de communications et de puissance de calcul de l'architecture proposée dans la figure III.4 indique que le nombre moyen de messages échangés entre les agents augmente linéairement avec le nombre de sources à gérer. De même, la durée moyenne de l'optimisation croît rapidement. L'analyse de ces premiers résultats indique que les exigences en matière de communication et de puissance de calcul sont limitées, et que la gestion de quelques dizaines de turbines est possible. Toutefois, le contrôle d'un grand nombre de sources augmenterait ces besoins rapidement (d'autant plus que le nombre d'itérations dans l'optimisation devra être augmenté), et un système de contrôle plus décentralisé deviendra alors plus approprié. C'est pour cette raison que nous travaillons sur la proposition d'une nouvelle approche qui se basera sur un système multi-agents complètement décentralisé. Dans un tel système, chaque agent devra être doté d'une intelligence supérieure à celle des agents qui composent l'architecture actuelle. Ce travail fait l'objet d'un challenge lancé en collaboration avec des collègues spécialistes des systèmes multi-agents. Un autre point important sur lequel nous continuerons à travailler porte sur l'agent (voir Figure III.4) qui sera chargé de la *planification* à l'avance, si possible optimale, du fonctionnement de l'ensemble, à l'aide de prévisions de production, de demande et de prix émis par un des agents. Le problème de la *planification* est une extension du problème appelé *dispatching*. Au lieu de considérer simplement les points de fonctionnement à un

instant t donné, on considère des profils de plusieurs heures (voire plusieurs jours). Cela permet de déterminer quand démarrer telle ou telle source, quand il est optimal de stocker de l'énergie en prévision d'un pic de demande, quand il est préférable de délester une partie de la charge, etc.

La résolution de ce problème passe, d'une part, par l'obtention et l'utilisation de prévisions, aussi précises que possible, de la production intermittente, de la demande et des prix d'achat ou revente d'énergie, et d'autre part, par le développement de nouvelles métaheuristiques hybrides. La solution d'un tel problème comportera des variables mixtes (entiers et réels) déterminant à la fois si une unité doit démarrer, s'arrêter ou se mettre en veille, et le point de fonctionnement optimal pour chaque source. D'autres contraintes, telles que les durées minimales de fonctionnement ou d'arrêt des sources thermiques, sont également à ajouter. La dimension du problème étant alors bien plus grande, il est également bien plus long à résoudre, d'où la nécessité d'avoir recours au calcul parallèle. Toutefois, il est résolu en parallèle avec celui du *dispatching*, et ne nécessite pas d'être résolu en temps réel (une à deux résolutions par jour peuvent suffire, suivant la précision des prévisions).

Du point de vue recherche, pour résoudre le problème de la planification, je m'intéresse actuellement à une nouvelle classe, d'approches qui combine des métaheuristiques avec des solveurs de Programme Linéaire (PL) comme CPLEX, qui, en retour, dans la mesure du possible, donnent des informations sur la qualité des solutions qu'elles obtiennent. En effet, bien que ces solveurs soient réputés pour leur efficacité, ils ne sont pas capables de résoudre des problèmes de grande taille ou même de fournir des solutions dans des temps raisonnables pour des problèmes difficiles, qui sont souvent NP-complets lorsqu'il s'agit des PL à variables en nombres entiers (PLNE).

Une première démarche consiste à s'intéresser aux relaxations entières des problèmes PLNE. Or, dans la pratique, cela s'est révélé souvent infructueux. D'autre part, les métaheuristiques permettent de proposer des solutions simples et relativement de bonne qualité pour des problèmes complexes. Néanmoins, lorsque la taille du problème est trop importante, les voisinages classiques utilisés deviennent aussi inexploitables dans des temps raisonnables. Il faudrait néanmoins pouvoir combiner les performances de ces solveurs avec celles des métaheuristiques basées sur la construction d'une solution réalisable à partir de solutions irréalisable, ou sur l'amélioration progressive d'une solution, afin d'obtenir des méthodes approximatives plus performantes. En effet, certaines similarités existent entre les méthodes arborescentes, telle que *Branch & Bound*, et les métaheuristiques qui se basent essentiellement sur une exploration itérative de l'espace des solutions possibles,

tandis que les méthodes arborescentes classiques font une énumération explicite (intelligente) de toutes les solutions. Il serait intéressant de réduire la zone d'exploration initiale, tout en garantissant une certaine qualité parmi les solutions retenues pour être examinées. Dans certains cas, cela revient à déterminer un ensemble dominant de solutions.

– Optimisation du moteur électrique

Nous avons démontré dans le chapitre V que nos algorithmes sont très utiles dans le cadre de la conception des moteurs électriques. Nous continuerons à travailler, dans la continuité des deux projets européens cités dans le chapitre V, sur le développement de nouveaux algorithmes d'optimisation (basés sur l'approche Pareto) pour minimiser les pertes d'un moteur électrique à très haute vitesse, en tenant compte des paramètres dimensionnels (dimension de l'entrefer, largeur des encoches), du choix optimal des matériaux et de la structure (type des encoches, etc.). Les fortes densités de puissance dans les moteurs à grande vitesse, mais aussi les phénomènes physiques (les courants de Foucault entre autres) rendent l'optimisation de ces dispositifs plus complexe. Cela demande une procédure plus puissante qu'une simple étude paramétrique. Les algorithmes stochastiques à développer représentent une bonne solution pour trouver un optimum global, mais ils nécessitent l'estimation de la fonction objectif pour un grand nombre de configurations. Ce grand nombre de simulations peut s'avérer très coûteux dans le cas des résolutions par la méthode des éléments finis. Pour optimiser les structures électromagnétiques, nous proposons de piloter directement l'outil de simulation par un algorithme d'optimisation. Appliquée sur un nombre restreint de paramètres, cette optimisation sera précédée par un tri automatique des paramètres significatifs. Cela pourra servir par exemple à réduire les pertes du rotor par l'optimisation des ouvertures des encoches et de la valeur de l'entrefer.

– Optimisation de la reconstruction de formes 3D

La modélisation de formes tridimensionnelles en utilisant les logiciels existants est un travail fastidieux, qui demande une forte expérience et beaucoup de temps. En comparaison, le dessin d'une forme tridimensionnelle sur une feuille est un travail très simple. La plupart d'entre nous ont un don naturel pour dessiner des croquis de voitures, d'arbres, etc. Malheureusement, les techniques existantes, qui permettent de reconstruire des formes à partir de croquis, sont très primaires. Elles ne permettent de modéliser que des formes très simples.

Dans ce travail, nous proposons de développer des méthodes de construction de formes tridimensionnelles symétriques, à partir d'un dessin bidimensionnel. Les données d'en-

trée sont un croquis constitué d'un ensemble de courbes correspondant à la projection orthogonale de la silhouette d'une forme tridimensionnelle. Nous supposons que la forme à reconstruire est symétrique par rapport à un plan. La principale difficulté de la modélisation 3D à partir de croquis 2D est qu'il existe une infinité de solutions. Plusieurs formes peuvent avoir exactement la même silhouette. De plus, la reconstruction nécessite de calculer un nombre très important de variables. Afin de trouver une solution unique, la reconstruction est modélisée sous la forme d'un problème d'optimisation avec une fonction objectif et un ensemble de contraintes. Une partie importante de ce travail concerne donc la conception d'algorithmes d'optimisation. Ce travail fait l'objet de la thèse de M. Chérin, qui a démarré le 01/10/2011, et vise les objectifs suivants :

- Trouver un modèle pour représenter la forme à reconstruire. Cette étape est importante ; ce modèle servira à formuler le problème d'optimisation. Il faudra trouver un bon compromis entre un modèle trop simple, qui limiterait la complexité des formes que l'on peut reconstruire, et un modèle trop compliqué, pour lequel le temps de calcul sera trop long.
- Formulation du problème d'optimisation. En fonction du modèle choisi, il faudra définir les variables. Ce problème d'optimisation sera formulé avec une fonction multiobjectif et un ensemble de contraintes. La fonction multiobjectif permettra, par exemple, de reconstruire la forme dont le ratio surface sur volume est le plus faible possible. Les contraintes seront nécessaires, afin que la silhouette de la forme reconstruite corresponde au dessin.
- Mise en œuvre des méthodes hybrides permettant de résoudre efficacement le problème multiobjectif lié à la reconstruction de formes 3D. Les métaheuristiques constituent une classe d'approches intéressante pour résoudre les problèmes de grande taille que nous aurons à traiter. En effet, leur variété permet de s'adapter à différents types de problèmes. Pour tirer avantage de cette variété, de nombreuses méthodes hybrides, mettant en œuvre deux métaheuristiques par exemple, existent dans la littérature. Malheureusement, actuellement, ce type d'hybridation est majoritairement réalisé de façon statique et est paramétré de manière expérimentale. Ainsi, l'une des limites de ce type de méthode concerne l'ensemble des paramètres d'hybridation à définir (comment combiner deux approches, à quel moment instancier telle ou telle approche, etc.). Nous souhaitons, à travers, cette thèse répondre à ces questions.

– **Développement de métaheuristiques massivement parallèles**

Nous avons récemment décroché un financement dans le cadre d'un projet BQR déposé

à l'UHA en 2012. Le but de ce financement est de nous permettre d'acheter douze cartes GPU (GTX680) afin de créer un *cluster* GPU à l'UHA. D'un point de vue scientifique, ce financement nous permettra d'expérimenter nos algorithmes d'optimisation sur des *clusters* hétérogènes (il est à noter que, jusqu'à présent, tous les supercalculateurs sont composés de machines homogènes) et de disposer localement d'une puissance de calcul phénoménale : cela va nous permettre de nous attaquer à des problèmes tels que la résolution structurale de nouvelles zéolithes³ (voir l'exemple de la figure VI.11), la gestion électrique dans les *smarts grids*, l'allocation de fréquences, etc.

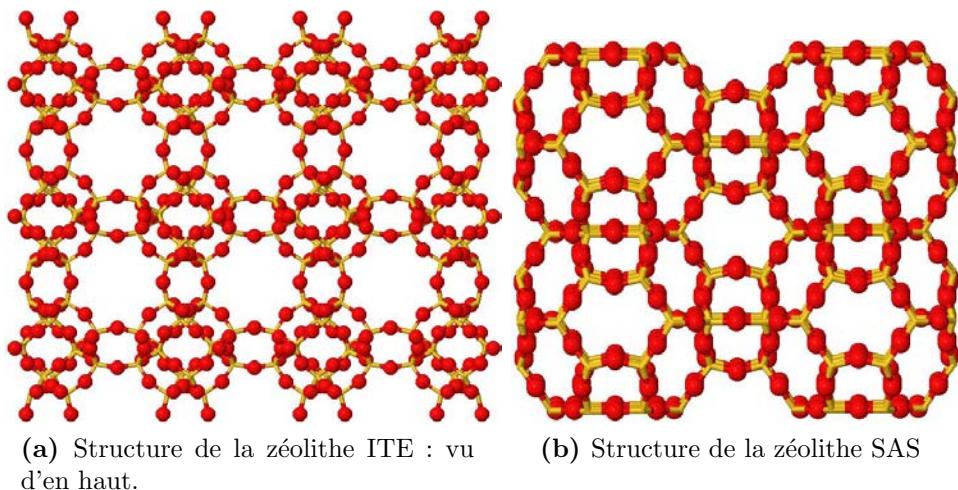


Figure VI.11 – Deux exemples de zéolithes.

En ce qui concerne notre collaboration avec l'équipe MPC de l'IS2M-UHA (Jean-Louis Paillaud) et l'équipe BFO de le LSiiT-UDS (Prof. P. Collet) nous avons recruté en 2012 un stagiaire pour développer une métaheuristique massivement parallèle pour la résolution de structures zéolithiques. Très brièvement, le principe des calculs est de proposer des modèles de structures simulés à partir de données telles que le paramètre de maille, la composition chimique et éventuellement le groupe d'espace et en utilisant le tétraèdre TO_4 comme brique élémentaire. Cette approche “prédition de structures” est prévue pour fournir des solutions candidates stables, sans comparer leurs données de diffraction théoriques à une expérimentation au cours du calcul et ce, dans un délai raisonnable, grâce

3. Les zéolithes sont formées de structures cristallines dans lesquelles quatre atomes d'oxygène enferment un atome de silicium ou d'aluminium. Elles sont utilisées pour filtrer (on les a qualifiées de tamis moléculaire), dans les adoucisseurs d'eau, les colonnes échangeuses d'ions, dans les systèmes de production d'oxygène ou encore dans la séparation des produits pétroliers ou lors de l'extraction du gaz naturel. Elles sont aussi utilisées comme catalyseurs dans de nombreux types de réactions chimiques. Comme transporteur de potassium, certaines font office d'engrais pour l'agriculture.

à la puissance des *clusters* GPU. Ensuite, les différentes solutions possibles proposées par l'algorithme sont affinées par une procédure basée sur la technique de potentiels interatomiques, en utilisant le code GULP ou d'autres champs de forces. Un premier exemple d'utilisation de cet algorithme a été publié très récemment dans le cas de la résolution structurale de la zéolithe ITQ-43.

– Optimisation de l'énergie dans un réseau de capteurs

La collaboration récente avec l'équipe GRTC du MIPS, à l'UHA, sur l'optimisation des réseaux de capteurs, amène également des perspectives d'application de nouvelles approches métaheuristiques d'optimisation distribuées. Ce travail, qui fait actuellement l'objet d'un sujet de stage de M2, a pour objectif de s'intéresser à la problématique de déploiement des réseaux de capteurs statiques (et dynamiques, par la suite, avec des obstacles qui se déplacent). Le but visé étant de générer la meilleure topologie du réseau, en prenant en compte plusieurs objectifs, tels que le coût du déploiement, la connectivité du réseau, la qualité de détection et la durée de vie du réseau. Le problème ainsi formalisé est un problème d'optimisation multiobjectif. Pour le résoudre, nous proposons d'étudier les approches de déploiement telles que “Differentiated Deployment Algorithm” [Aits 07] et “Potential Field Deployment Algorithm” [NAit 11], ensuite de les combiner avec les approches dites Pareto. L'algorithme d'optimisation qui sera développé sera couplé avec le simulateur ns-2, qui permet la description et la simulation de réseaux de capteurs, dans le but de valider nos résultats.

– Evaluation des performances des métahéuristiques

Les performances que j'ai présentées dans ce mémoire ont été obtenues, pour l'essentiel, par des évaluations statistiques. Parmi les approches auxquelles nous nous sommes intéressées ces dernières années, on trouve les algorithmes génétiques et les processus décisionnels de Markov (non présentés dans ce mémoire) qui sont modélisables en termes de chaîne de Markov dont la distribution stationnaire représente l'optimum à atteindre. Les algorithmes itératifs correspondants possèdent la propriété dite de “contraction” et il est possible d'appliquer le théorème du point fixe. On obtient ainsi un majorant pour la vitesse de convergence vers l'optimum (inégalité de Bellman). C'est ce que j'ai fait dans le travail concernant l'allocation des fréquences⁴.

La modélisation et l'analyse des métahéuristiques est beaucoup plus difficile et nécessite

4. voir la liste de mes publications pour les références.

la mise au point de nouvelles méthodes mathématiques d'évaluation des performances. J'ai entrepris un travail dans ce sens avec R. Schott et ses collaborateurs. Notre objectif est de prouver des résultats mathématiques concernant les performances en moyenne, les performances garanties, la robustesse et la qualité des solutions et de les comparer avec ceux obtenus statistiquement. Il s'agit là de recherches en cours qui n'ont pas encore fait l'objet de publications.

Références bibliographiques

Références bibliographiques

- [Absa 05] J. Absar and F. Catthoor. “Compiler-Based Approach for Exploiting Scratch-Pad in Presence of Irregular Array Access”. In : I. C. Society, Ed., *Proceedings of the conference on Design, Automation and Test in Europe*, pp. 1162 – 1167, IEEE, Munich, Germany, March 7- 11 2005. [80](#)
- [Absa 06] J. Absar and F. Catthoor. “Analysis of Scratch-Pad and Data-Cache Performance Using Statistical Methods”. In : I. C. Society, Ed., *Proceedings of the 2006 Conference on Asia South Pacific Design*, pp. 820–825, IEEE, Yokohama, Japan, January 24-27 2006. [80](#)
- [Aits 07] N. Aitsaadi, N. Achir, kh. Boussetta, and G. Pujolle. “Differentiated Underwater Sensor Network Deployment”. In : *Proceeding of the IEEE/OES OCEANS*, pp. 1–6, Aberdeen, Scotland, June 18-21 2007. [136](#)
- [Al k 08] A. Al-khedhairi. “Simulated Annealing Metaheuristic for Solving P-Median Problem”. *Contemporary Mathematical Sciences*, Vol. 3, No. 28, pp. 1357–1365, 2008. [15](#)
- [Alba 05] E. Alba. *Parallel Metaheuristics : A New Class of Algorithms*. John Wiley & sons, 2005. [12](#)
- [Angi 03] E. Angiolini, L. Benini, and A. Caprara. “Polynomial-Time Algorithm for On-Chip Scratchpad Memory Partitioning”. In : *Proceedings of the International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, pp. 318–326, ACM, San Jose, California, USA, October 30 - November 1 2003. [80](#)
- [Atas 07] E. Atashpaz-Gargari and C. Lucas. “Imperialist competitive algorithm : An algorithm for optimization inspired by imperialistic competition”. In : *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 4661 – 4667, Singapore, September 25 - 28 2007. [68](#)

Références bibliographiques

- [Avis 02] O. Avissar, R. Barua, and D. Stewart. “An Optimal Memory Allocation Scheme for Scratch-pad-based Embedded Systems”. *Transactions on Embedded Computing Systems*, Vol. 1, No. 1, pp. 6–26, 2002. [85](#)
- [Azen 92] R. Azencott. *Simulated Annealing : Parallelization Techniques*. John Wiley & sons, 1992. [14](#)
- [Bana 02] R. Banakar, S. Steinke, B. Lee, M. Balakrishnan, and P. Marwedel. “Scratchpad Memory : Design Alternative for Cache On-Chip Memory in Embedded Systems”. In : *Proceedings of the tenth international symposium on Hardware/software codesign*, pp. 73–78, ACM, Estes Park, Colorado, USA, 2002. [80](#)
- [Bari 03] V. Barichard. *Approches hybrides pour les problèmes multiobjectifs*. PhD thesis, Université d’Angers, 2003. [9](#), [26](#)
- [Baum 04] U. Baumgartner, C. Magele, and W. Renhart. “Pareto optimality and particle swarm optimization”. *IEEE Transactions on Magnetics*, Vol. 40, No. 2, pp. 1172–1175, 2004. [27](#)
- [Bddo 96] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University, Press, 1996. [113](#)
- [Beas 93] J. Beasley. *Langrangean relaxation*. C.R. Reeves (Ed.) Modern heuristics for combinatorial problems, Blackwell Scientific Publishing, Oxford, 1993. [9](#)
- [Ben 05] H. Ben Fradj, A. El Ouadihi, C. Belleudy, and M. Auguin. “Energy Aware Memory Architecture Configuration”. *ACM SIGARCH Computer Architecture News*, Vol. 33, No. 3, pp. 3–9, September 2005. [79](#), [80](#), [81](#)
- [Benc 10] Benchmarks. 2010. www.loria.fr/~idrissma/benchs.zip. [84](#)
- [Beni 00] L. Benini and G. D. Micheli. “System-Level Power Optimization : Techniques and Tools”. *ACM Transactions on Design Automation of Electronic Systems*, Vol. 5, No. 2, pp. 115–192, 2000. [80](#)
- [Berr 01] A. Berro. *Optimisation multiobjectif et stratégies d’volution en environnement dynamique*. PhD thesis, Thèse de doctorat. Université des Sciences Sociales Toulouse 1, December 2001. [28](#)
- [Bran 02] J. Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, 2002. [11](#)
- [Bras 02] D. Brash. “The ARM Architecture Version 6 (ARMv6)”. January 2002. [80](#)
- [Chao 06] D. Chaojun and Z. Qiu. “Particle Swarm Optimization Algorithm Based on the Idea of Simulated Annealing”. *Computer Science and Network Security*, Vol. 6, No. 10, pp. 152–157, October 2006. [88](#)

- [Chow 04] C. Chow and H. Tsui. “Autonomous agents response learning by multispecies particle swarm optimization”. In : *Proceeding of the IEEE Congress on Evolutionary Computation*, pp. 778–785, Portland OR, USA, June 19-23 2004. [28](#)
- [Cler 05] M. Clerc. *L'optimisation par Essaims Particulaires*. Hermès Science Publications, 2005. [23](#)
- [Coel 02a] C. C. Coello and M. S. Lechuga. “MOPSO : a proposal for multiple objective particle swarm optimization”. In : *Proceeding of the 2002 IEEE Congress on Evolutionary Computation*, pp. 1051–1056, Honolulu, Hawaii, May 12-17 2002. [29](#)
- [Coel 02b] C. C. Coello, D. Veldhuizen, and G. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002. [29](#)
- [Coel 04] C. C. Coello, G. T. Pulido, and M. S. Lechuga. “Handling multiple objectives with particle swarm optimization”. *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 256–279, June 2004. [29](#)
- [Coll 02] Y. Collette and P. Siarry. *Optimisation Multiobjectif*. Eyrolles, 2002. [11](#), [12](#), [26](#), [27](#), [30](#)
- [Colo 91] A. Colorni, M. Dorigo, and V. MANIEZZO. “Distributed optimization by ant colonies”. In : *Proceedings of the European Conference on Artificial Life*, pp. 134–142, Paris, France, November 20-21 1991. [19](#)
- [Come 98] F. Comellas and J. Ozón. “An ant algorithm for the graph Coloring problem”. In : *Proceeding of the First International Workshop on Ant Colony Optimisation*, Brussels, Belgium, October 15-16 1998. [39](#), [43](#)
- [Coor 08] Y. Cooren. *Perfectionnement d'un algorithme adaptatif d'Optimisation par Esaim Particulaire. Applications en génie médical et en électronique*. PhD thesis, Université Paris 12 - Val De Marne, 2008. [26](#)
- [Coup 99] J.-M. Coupé. “Utilisation de la méthode tabou pour le problème d’allocation de fréquences”. Brevet TDF N. 99 02768, 1999. [32](#), [37](#), [39](#), [43](#)
- [Crep 05] J. Créput, A. Koukam, T. Lissajoux, and A. Caminada. “Automatic Mesh Generation for Mobile Network Dimensioning using Evolutionary Approach”. *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 1, pp. 18–30, 2005. [32](#)
- [Crep 06] J. Créput and A. Koukam. “Local search study of honeycomb clustering problem for cellular planning”. *Mobile Network Design and Innovation*, Vol. 1, No. 2, pp. 153–160, 2006. [32](#)
- [Crep 08] J. Créput. *Hybridation de métaheuristiques pour la résolution distribuée de problèmes d'optimisation spatialisés*. PhD thesis, Université de Bourgogne, 2008. [11](#), [26](#)

Références bibliographiques

- [Crom 94] W. Crompton, S. Hurley, and N. Stephens. “A parallel genetic algorithm for frequency assignment problems”. In : *Proceedings of the IMACS/IEEE Conference on Signal Processing Robotics and Neural Networks*, pp. 81–84, Lille, France, April 25-27 1994. [32](#)
- [Deb 01] K. Deb. *Multiobjective optimization using evolutionary algorithms*. John Wiley & sons, 2001. [26](#)
- [Deb 02] K. Deb. “A fast elitist non-dominated sorting genetic algorithm for multiobjective optimization : NSGA II”. *IEEE Transactions on Evolutionary computation*, Vol. 5, No. 3, pp. 115 – 148, 2002. [29](#)
- [Deb 08] K. Deb and S. Tiwari. “Omni-Optimizer : A Generic Evolutionary Algorithm for Single and Multi-objective Optimization”. *European Journal of Operational Research*, Vol. 185, No. 3, pp. 1062–1087, 2008. [11](#)
- [Deb 95] K. Deb and R. B. Agrawal. “Simulated binary crossover for continuous search space”. *Complex Systems*, Vol. 9, No. 2, pp. 115–148, 1995. [104](#)
- [Deb 96] K. Deb and M. Goyal. “A combined genetic adaptive search (geneas) for engineering design”. *Computer Science and Informatics*, Vol. 26, No. 4, pp. 30–45, 1996. [104](#)
- [Dhae 05] C. Dhaenens-Flipo. *Optimisation Combinatoire Multi-Objectif : Apport des Méthodes Coopératives et Contribution à l'Extraction de Connaissances*. PhD thesis, Université des Sciences et Technologies de Lille, 2005. [9](#), [23](#), [26](#)
- [Domi 05] A. Dominguez, S. Udayakumaran, and R. Barua. “Heap Data Allocation to Scratch-Pad Memory in Embedded Systems”. *ACM Transactions on Design Automation of Electronic Systems*, Vol. 5, No. 2, pp. 115–192, 2005. [85](#)
- [Dori 96] M. Dorigo, V. Maniezo, and A. Colorni. “The ant system : optimization by a colony of cooperating agents”. *IEEE Transaction on Systems, Man and Cybernetics*, Vol. 26, No. 1, pp. 1–13, 1996. [20](#)
- [Dori 97a] M. Dorigo and L. Gambardella. “Ant colonies for the traveling salesman Problem”. Tech. Rep., Université Libre de Bruxelles, 1997. [20](#)
- [Dori 97b] M. Dorigo and L. Gambardella. “Ant colony system : a cooperative learning approach to the traveling salesman problem”. *IEEE Transactions on Evolutionary Computing*, Vol. 1, pp. 53–66, 1997. [21](#)
- [Dorn 95] R. Dorne and J. Hao. “An Evolutionary Approach for Frequency Assignment in Cellular Radio Networks”. In : *Proceeding of the IEEE International Conference on Evolutionary Computation*, pp. 539 – 544, IEEE, Perth, Western Australia, November 29 - December 1 1995. [32](#)

- [DWol 96] D.Wolpert and W. Macready. “No free lunch theorems for optimization”. *IEEE Transactions on Evolutionnary Computation*, Vol. 1, pp. 67–82, 1996. [11](#)
- [Ehrg 00a] M. Ehrgott. *Multicriteria optimization. Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, 2000. [26](#)
- [Ehrg 00b] M. Ehrgott and X. Gandibleux. “A survey and annotated bibliography of multiobjective combinatorial optimization”. *Operations Research Spectrum*, Vol. 22, pp. 425–460, 2000. [26](#)
- [Engr 97] P. Engrand. “Approche d’optimisation multiobjective basée sur l’adoucissement simulé, et son application à la gestion des combustibles nucléaires”. In : *Proceedings of the Fifth International Conference on Nuclear Engineering*, pp. 1 – 8, Nice, France, May 26 - 30 1997. [27](#)
- [Fang 07] L. Fang, P. Chen, and S. Liu. “Particle Swarm Optimization with Simulated Annealing for TSP”. In : *Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pp. 206 – 210, World Scientific and Engineering Academy and Society (WSEAS), Corfu Island, Greece, February 16 - 19 2007. [88](#)
- [Feok 06] V. Feoktistov. *Differential Evolution : In Search of Solutions (Springer Optimization and Its Applications)*. Vol. 5, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. [124](#)
- [Ferb 99] J. Ferber. *Multi-Agent Systems : An Introduction to Artificial Intelligence*. Addison-Wesley, 1999. [59](#)
- [Fodo 09] D. Fodorean and A. Miraoui. “Dimensionnement rapide des machines synchrones à aimants permanents (MSAP)”. 10 novembre 2009. <http://www.techniques-ingenieur.fr/res/pdf/encyclopedia/42249210-d3554.pdf>. [98](#)
- [Fodo 12] D. Fodorean, L. Idoumghar, and L. Szabo. “Motorization for electric scooter by using permanent magnet machines optimized based on hybrid metaheuristic algorithm”. *IEEE Transaction on Vehicular Technology*, July 2012. Accepted for publication. [97](#)
- [Fons 95] C. M. Fonseca and P. J. Fleming. “An overview of evolutionary algorithms in multiobjective optimization”. *Evolutionary computation*, Vol. 3, No. 1, pp. 1 – 16, 1995. [29](#)
- [Frie 93] T. L. Friesz. “The multiobjective equilibrium network design problem revisited : A simulated annealing approach”. *European Journal of Operational Research*, Vol. 65, pp. 44 – 57, 1993. [27](#)

- [G Di 98] M. D. G. Di Caro. “Ant colonies for adaptive routing in packetswitched communications networks”. In : *Proceeding of the 5th International Conference on Parallel Problem Solving from Nature*, pp. 673–682, Springer, Amsterdam, The Netherlands, September 27-30 1998. [11](#)
- [Gamb 97] L. Gambardella, E. Taillard, and M. Dorigo. “Ant colonies for the quadratic assignment problem”. Rapport technique IDSIA-4-97, IDSIA, Lugano, 1997. [21](#)
- [Garc 09] S. Garcìà, M. L. D. Molina, and F. Herrera. “A study on the use of non-parametric tests for analyzing the evolutionary algorithms’s behaviour : a case study on the CEC’2005 special session on real parameter optimization”. *Journal of Heuristics*, Vol. 15, pp. 617 – 644, 2009. [6](#), [119](#)
- [Gend 03] M. Gendreau. *An Introduction to Tabu Search*. Vol. 57, Kluwer Academic Publishers, Boston, MA, 2003. [15](#)
- [Giur 08] S. Giurgea, D. Fodorean, G. Cirrincione, A. Miraoui, and M. Cirrincione. “Multimodel Optimization Based on the Response Surface of the Reduced FEM Simulation Model With Application to a PMSM”. *IEEE Transactions on Magnetic*, Vol. 44, No. 2, pp. 2153–2157, 2008. [98](#), [102](#), [106](#)
- [Glov 65] F. Glover. “A multiphase-dual algorithm for the zero-one integer programming problem”. *Operations Research*, Vol. 13, pp. 879–919, 1965. [9](#)
- [GLOV 77] F. GLOVER. “Heuristics for integer programming using surrogate constraints”. *Decision Sciences*, Vol. 8, pp. 156 – 166, 1977. [9](#)
- [Glov 97] F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, 1997. [34](#)
- [Gold 87] D. Goldberg and J. Richardson. “Genetic algorithms with sharing for multimodal function optimisation”. In : *Proceeding of the Second International Conference on Genetic Algorithms*, pp. 41 – 49, L. Erlbaum Associates Inc., Cambridge, Massachusetts, USA, July 28 - 31 1987. [11](#)
- [Gold 89] D. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989. [34](#)
- [Gray 02] R. Graybill and R. Melhem. *Power Aware Computing*. Kluwer Academic Publishers, Norwell, MA, USA, 2002. [78](#)
- [Guth 01] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown. “MiBench : A free, Commercially Representative Embedded Benchmark Suite”. In : *Proceedings of the Workload Characterization. WWC-4 IEEE International Workshop*, pp. 3 – 14, IEEE, Austin, Texas, USA, December 2 2001. [84](#)
- [H Ke 04] U. P. H. Kellerer and D. Pisinger. *Knapsack Problems*. Springer, Berlin, Germany, 2004. [83](#)

- [hadj 08] N. B. hadj, S. Tounsi, R. Neji, and F. Sellami. “Use of Evolutionary Approach for Multiobjective Optimization of Losses and Mass in Permanent Magnet Motor”. *Electrical Systems*, Vol. 4, No. 4, pp. 1–16, December 2008. [98](#)
- [Hao 99] J. Hao, P. Galinier, and M. Habib. “Métaheuristiques pour l’optimisation combinatoire et l’affectation sous contraintes”. *Revue d’Intelligence Artificielle*, Vol. 13, No. 2, pp. 283 – 324, 1999. [9](#)
- [Help 70] M. Help and R. M. Karp. “The travelling-saleman problem and minimum spanning trees”. *Operations Research*, Vol. 18, pp. 1138 – 1162, 1970. [9](#)
- [Hila 08] V. Hilaire, A. Koukam, and S. Rodriguez. “An Adaptative Agent Architecture for Holonic Multi-Agent Systems”. *ACM Transactions on Autonomous and Adaptive Systems*, Vol. 3, No. 1, pp. 1–24, 2008. [59](#)
- [Holl 92] J. Holland. “Les algorithmes génétiques”. *Pour la Science*, Vol. 179, pp. 44 – 51, 1992. [37](#)
- [Hu 02] X. Hu and R. Eberhart. “Multiobjective optimization using dynamic neighborhood particle swarm optimization”. In : *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pp. 1677 – 1681, Honolulu, Hawaii, May 12 - 17 2002. [28](#)
- [Hu 03] X. Hu, R. Eberhart, and Y. Shi. “Particle Swarm with Extended Memory for Multiobjective Optimization”. In : *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, pp. 193 – 197, West Lafayette, IN, USA, April 24 - 26 2003. [28](#)
- [Hurl 01] S. Hurley. “Planning effective cellular mobile radio networks”. *IEEE Transactions on Vehicular Technology*, Vol. 51, No. 2, pp. 443–472, 2001. [32](#)
- [idou 01a] idoumghar. “Procédé d’allocation de fréquences à un ensemble d’émetteurs / récepteurs en réseau par un processus génétique hybride”. Tech. Rep. 01 06885, May 2001. TDF-C2R. [31](#)
- [Idou 01b] Idoumghar, M. Alabau, and R. Schott. “A New Hybrid Genetic Algorithm for the Graph Colouring Problem”. In : *Proceedings of the 3rd Colloquium on Computational Telecommunications*, pp. 111–117, Saint Jean de Luz, France, May 28-30 2001. [31](#)
- [Idou 01c] L. Idoumghar, M. Alabau, and R. Schott. “New Hybrid Genetic Algorithms for The Frequency Assignment Problem”. In : *Proceedings of the 13th International IEEE Conference on Tools with Artificial Intelligence*, pp. 136 – 142, IEEE, Dallas, Texas, USA, November 7 - 9 2001. [31](#)
- [Idou 02a] L. Idoumghar. *Méthodes Algorithmiques pour l’Allocation de Fréquences*. PhD thesis, Université Henri Poincaré, December 12 2002. [37](#)

Références bibliographiques

- [Idou 02b] L. Idoumghar, M. Alabau, and R. Schott. “New Hybrid Genetic Algorithms for The Frequency Assignment Problem”. *IEEE Transactions on Broadcasting*, Vol. 48, No. 1, pp. 27 – 34, March 2002. [12](#), [31](#), [34](#)
- [Idou 02c] L. Idoumghar, M. Alabau, and R. Schott. “Nouvelle approche hybride pour résoudre le problème d’allocation de fréquences”. In : *Proceedings of Reconnaissance des Formes et Intelligence Artificielle*, pp. 241–248, AFRIF-AFIA Pub, Angers, France, January 8-10 2002. [31](#)
- [Idou 02d] L. Idoumghar and P. Debreux. “New Modeling Approach for the Frequency Assignment Problem in Broadcasting”. *IEEE Transactions on Broadcasting*, Vol. 48, No. 4, pp. 293–298, December 2002. [31](#), [50](#)
- [Idou 02e] L. Idoumghar and P. Debreux. “New Modelling Approach for the Frequency Assignment Problem in Broadcasting”. In : *Proceedings of the IASTED International Conference on Software Engineering and Applications*, pp. 805 – 812, M.H. Hamza Ed, Combridge, USA, November 4 - 6 2002. [31](#), [50](#)
- [Idou 02f] L. Idoumghar, R. Schott, and J. Greff. “Using Markov Decision Processes to Solve the Frequency Assignment Problem”. In : *Proceedings of the 6th IASTED International Conference on Artificial intelligence and Soft Computing*, pp. 552 – 557, Acta, Banff, Canada, July 17-19 2002. [31](#)
- [Idou 03] L. Idoumghar and R. Schott. “Probabilistic and Algorithmic Methods for the Frequency Assignment Problem”. In : *Proceedings of the 5th International Congress on Industrial and Applied Mathematics*, pp. 228 – 229, Sydney, Australia, July 7 - 11 2003. [31](#)
- [Idou 04] L. Idoumghar, R. Schott, and J. Greff. “Application of Markov Decision Processes to the Frequency Assignment Problem”. *Applied Artificial Intelligence*, Vol. 18, No. 8, pp. 761–773, September 2004. [31](#)
- [Idou 06] L. Idoumghar and R. Schott. “A New Hybrid GA-MDP Algorithm For The Frequency Assignment Problem”. In : *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pp. 18 – 27, IEEE, Washington D.C, USA, November 13 - 15 2006. [31](#)
- [Idou 09a] L. Idoumghar, M. Melkemi, and R. Schott. “A Novel Hybrid Evolutionary Algorithm for Multi-Modal Function Optimization and Engineering Applications”. In : *Proceedings of the 13th IASTED International Conference on Artificial Intelligence and Soft Computing*, pp. 87 – 93, Acta, Palma de Mallorca, Spain, September 7 - 9 2009. [24](#)
- [Idou 09b] L. Idoumghar, T. Raminosoa, and A. Miraoui. “New Tabu Search Algorithm to Design an Electric Motor”. *IEEE Transactions on Magnetic*, Vol. 45, No. 3, pp. 1498 – 1501, March 2009. [97](#)

- [Idou 09c] L. Idoumghar and R. Schott. “Two Distributed Algorithms for the Frequency Assignment Problem in the Field of RadioBroadcasting”. *IEEE Transactions on Broadcasting*, Vol. 55, No. 2, pp. 223 – 229, June 2009. [31](#), [34](#)
- [Idou 10a] L. Idoumghar, D. Fodorean, and A. Miraoui. “Using Hybrid Constricted Particles Swarm and Simulated Annealing Algorithm for Electric Motor Design”. In : *Proceedings of the 14th Biennial IEEE Conference on Electromagnetic Field Computation*, Chicago - Illinois USA, May 9 - 12 2010. CD-ROM. [97](#)
- [Idou 10b] L. Idoumghar, M. Idrissi-Aouad, M. Melkemi, and R. Schott. “Metropolis Particle Swarm Optimization Algorithm with Mutation Operator For Global Optimization Problems”. In : *Proceedings of the 22th International Conference on Tools with Artificial Intelligence*, pp. 35 – 42, IEEE, Arras, France, October 27 - 29 2010. [65](#)
- [Idou 10c] L. Idoumghar, M. Idrissi-Aouad, and R. Schott. “Sequential and Cooperative Distributed SA-Type Algorithms for Energy Optimization in Embedded Systems”. In : *Proceedings of the 2010 International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, Wuhan, China, December 10-12 2010. [77](#)
- [Idou 10d] L. Idoumghar, M. Idrissi-Aouad, R. Schott, and O. Zendra. “Reduction of Energy Consumption in Embedded Systems : A Hybrid Evolutionary Algorithm”. In : *Proceedings of the 3rd International Conference on Metaheuristics and Nature Inspired Computing*, Djerba, Tunisia, October 27-31 2010. [77](#), [88](#), [107](#)
- [Idou 10e] L. Idoumghar, M. Idrissi-Aouad, R. Schott, and O. Zendra. “Sequential and Distributed Hybrid GA-SA Algorithms for Energy Optimization in Embedded Systems”. In : *Proceedings of IADIS International Conference Applied Computing*, pp. 167–174, Timisoara, Romania, October 14-16 2010. [77](#), [86](#), [93](#)
- [Idou 10f] L. Idoumghar, M. M. M. Idrissi-Aouad, and R. Schott. “Metropolis Particle Swarm Optimization Algorithm with Mutation Operator For Global Optimization Problems”. In : *Proceedings of 22th IEEE International Conference on Tools with Artificial Intelligence*, pp. 35–42, Arras, France, October 27-29 2010. [77](#), [107](#)
- [Idou 11a] L. Idoumghar, D. Fodorean, A. N’diaye, D. Bouquain, and A. Miraoui. “Simulated Annealing Algorithm for the Optimization of an Electrical Machine”. *IET Electric Power Applications*, 2011. Accepted for publication. [97](#), [102](#), [103](#), [107](#)
- [Idou 11b] L. Idoumghar, M. Idrissi-Aouad, M. Melkemi, and R. Schott. “Hybrid PSO-SA Type Algorithms for Multi-Modal Function Optimization and Reducing Energy Consumption in Embedded Systems”. *Applied Computational Intelligence and Soft Computing*, Vol. 2011, pp. 1 – 12, June 2011. [77](#), [107](#)
- [Idou 12] L. Idoumghar, R. Roche, B. Blunier, and A. Miraoui. “Algorithmes hybrides pour la gestion intelligente de l’énergie dans les smart grids”. In : *Proceedings of 7èmes Journées Francophones Planification, Décision, et Apprentissage pour la conduite de systèmes*, pp. 1–12, Nancy, France, May 22 - 23 2012. [57](#)

Références bibliographiques

- [Idri 07] M. Idrissi Aouad and O. Zendra. “A Survey of Scratch-Pad Memory Management Techniques for low-power and -energy”. In : *Proceedings of the 2nd ECOOP Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems*, pp. 31–38, Berlin, Germany, July 30 - 31 2007. [80](#)
- [Idri 10] M. Idrissi Aouad, R. Schott, and O. Zendra. “A Tabu Search Heuristic for Scratch-Pad Memory Management”. In : *Proceedings of International Conference on Software Engineering and Technology*, pp. 386 – 390, WASET Publisher, Rome, Italy, April 28 - 30 2010. [86](#), [87](#), [91](#)
- [Idri 11] M. Idrissi-Aouad. *Conception d’algorithmes hybrides pour l’optimisation de l’énergie Mémoire dans les systèmes embarqués et de fonctions multimodales*. PhD thesis, Université Henri Poincaré - Nancy 1, 2011. [77](#)
- [Ishi 98] H. Ishibuchi and T. Murata. “A multiobjective genetic local search algorithm and its application to flowshop scheduling”. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 28, No. 3, pp. 1732 – 1743, 1998. [27](#)
- [ITRS 07] ITRS. “System Drivers”. 2007. http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_SystemDrivers.pdf. [78](#), [79](#)
- [Jin 01] Y. Jin, T. Okabe, and B. Sendho. “Dynamic weighted aggregation for evolutionary multiobjective optimization : why does it work and how ?”. In : *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1042 – 1049, San Francisco, California, USA, July 7 - 11 2001. [27](#)
- [Jour 10] L. Jourdan. *Métaheuristiques Coopératives : du déterministe au stochastique*. PhD thesis, Université Lille I, 2010. [26](#)
- [Kand 02] M. T. Kandemir, J. Ramanujam, and A. N. Choudhary. “Exploiting shared scratch pad memory space in embedded multiprocessor systems”. In : *Proceedings of the 39th annual Design Automation Conference*, pp. 219–224, New Orleans, LA, USA, June 10 - 14 2002. [80](#)
- [Kenn 95] J. Kennedy and R. C. Eberhart. “Particle Swarm Optimization”. In : *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, Western Australia, November 27 - December 1 1995. [21](#)
- [Kirk 83] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. “Optimization by Simulated Annealing”. *Science*, Vol. 220, pp. 671–680, 1983. [13](#)
- [Kroh 05] R. A. Krohling. “Gaussian Particle Swarm with Jumps”. In : *Proceedings of the IEEE Congress on Evolutionary Computat.*, pp. 1226 – 1231, Dortmund, Germany, September 2 - 5 2005. [25](#), [122](#)

- [Laza 08] M. Lazarova. “Parallel Simulated Annealing for Solving the Room Assignment Problem on Shared and Distributed Memory Platforms”. In : *Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*, pp. 13 – 18, ACM, Gabrovo, Bulgaria, June 12 - 13 2008. [15](#)
- [LCTE 03] LCTES. “Compilation Challenges for Network Processors”. In : *Proceedings of the ACM Conference on Languages, Compilers and Tools for Embedded Systems*, ACM, San Diego, California, USA, June 11-13 2003. Slide. [80](#)
- [Li 08] H. Li and D. L. Silva. “Evolutionary Multi-objective Simulated Annealing with Adaptive and Competitive Search Direction”. In : *IEEE Congress on Evolutionary Computation*, pp. 3311 – 3318, IEEE, Hong Kong, China, June 1 - 6 2008. [15](#)
- [Lian 06] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar. “Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions”. *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, pp. 281–295, 2006. [122](#)
- [Mats 08] S. Matsui and S. Yamada. “Optimizing Hierarchical Menus by Genetic Algorithm and Simulated Annealing”. In : *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pp. 1587 – 1594, ACM, Atlanta, GA, USA, July 12-16 2008. [15](#)
- [Mazi 11] B. Mazigh, V. Hilaire, and A. Koukam. “Formal Specification of Holonic Multi-Agent Systems : Application to Distributed Maintenance Company”. In : *Proceedings of the 9th International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 135 – 140, Salamanca, Spain, April 6 - 8 2011. [59](#)
- [Melk 09] L. I. M. Melkemi and R. Schott. “A Novel Hybrid Evolutionary Algorithm for Multi-modal Function Optimization and Engineering Applications”. In : *Proceedings of the 13th IASTED, International Conference on Artificial Intelligence and Soft Computing*, pp. 87–93, Palma de Mallorca, Spain, September 7 - 9 2009. [107](#)
- [Metr 53] N. Metropolis, M. Rosenbluth, A. Teller, and E. Teller. “Equation of state calculation by fast computing machines”. *Chemical Physics*, Vol. 21, No. 6, pp. 1087–1092, 1953. [13](#)
- [Meun 02] H. Meunier. *Algorithmes évolutionnaires parallèles pour l'optimisation multiobjectif de réseaux de télécommunications Mobiles*. PhD thesis, Université des Sciences et Technologies de Lille, 2002. [26](#)
- [Miet 99] K. Miettinen. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, 1999. [26](#)
- [Molg 05] M. Molga and C. Smutnicki. “Test Functions for Optimization Needs”. 2005. Available at <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>. [108](#)

- [NAit 11] N.Aitsaadi, N. Achir, kh. Bousetta, and G. Pujolle. “Artificial potential field approach in WSN deployment : Cost, QoM, connectivity, and lifetime constraints”. *Computer Networks*, Vol. 55, No. 1, pp. 84–105, 2011. [136](#)
- [Naka 08] S. Nakano, A. Ishigame, and K. Yasuda. “Consideration of Particle Swarm Optimization Combined with Tabu Search”. *IEE Transactions on Electronics, Information and Systems*, Vol. 128, No. 7, pp. 1162–1167, July 2008. [24, 115, 116](#)
- [Naki 07] A. Nakib. *Conception de métaheuristiques d'optimisation pour la segmentation d'images. Application à l' $\hat{\omega}_2^1$ des images biomédicales.* PhD thesis, Université Paris 12 - Val De Marne, 2007. [26](#)
- [Netw 02] T. J. Network, M. Adiletta, M. Rosenbluth, D. Bernstein, G. Wolrich, and H. Wilkinson. “The Next Generation of Intel IXP Network Processors”. *Intel Technology Journal*, Vol. 6, No. 3, pp. 6–18, August 2002. [80](#)
- [Nguy 05] N. Nguyen, A. Dominguez, and R. Barua. “Memory Allocation for Embedded Systems with a Compile-Time-Unknown Scratch-Pad Size”. In : *Proceedings of the 2005 international conference on Compilers, architectures and synthesis for embedded systems*, pp. 115 – 125, ACM, San Francisco, California, USA, September 24-27 2005. [80](#)
- [Nguy 07] N. Nguyen, A. Dominguez, and R. Barua. “Scratch-Pad Memory Allocation without Compiler Support for Java Applications”. In : *Proceedings of the 2007 international conference on Compilers, architectures and synthesis for embedded systems*, pp. 85–94, ACM, Salzburg, Austria, September 30 - October 3 2007. [80](#)
- [Orti 05] D. Ortiz-Boyer, C. Herbaś-Martínez, and N. Garcíá-Pedrajas. “CIXL2 - A crossover operator for evolutionary algorithms based on population features”. *Artifical Intelligence Research*, Vol. 24, pp. 1–48, 2005. [111](#)
- [Ourique 02] C. Ourique, E. Biscaia, and J. Pinto. “The use of particle swarm optimization for dynamical analysis in chemical processes”. *Computers & Chemical Engineering*, Vol. 26, No. 12, pp. 1783 – 1793, 2002. [11](#)
- [Pand 97] P. R. Panda, N. Dutt, and A. Nicolau. “Efficient Utilization of Scratch-Pad Memory in Embedded Processor Applications”. In : *Proceedings of the 1997 European conference on Design and Test*, p. 7, IEEE, Paris, France, March 17 - 20 1997. [85](#)
- [Pant 08] M. Pant, R. Thangaraj, and A. Abraham. “Particle Swarm Based Meta-Heuristics for Function Optimization and Engineering Applications”. In : *Proceedings of the 7th Computer Information Systems and Industrial Management Applications*, pp. 84 – 90, Ostrava, Czech Republic, June 26 - 28 2008. [24, 115, 116](#)
- [Pars 02] K. Parsopoulos and M. Vrahatis. “Particle Swarm Optimization in Multiobjective Problems”. In : *Proceeding of the 2002 ACM Symposium on Applied Computing*, pp. 603–607, Madrid, Spain, March 11 - 14 2002. [27](#)

- [Pars 04] K. E. Parsopoulos, D. Tasoulis, and M. N. Vrahatis. “Multiobjective optimization using parallel vector evaluated particle swarm optimization”. In : *Proceeding of the IASTED International Conference on Artificial Intelligence and Applications*, pp. 823–828, Innsbruck, Austria, February 16 – 18 2004. [28](#)
- [Prem 10] K. Premalatha and A. M. Natarajan. “Combined Heuristic Optimization Techniques for Global Minimization”. *Advances in Soft Computing and Its Applications*, Vol. 2, No. 1, pp. 85–99, March 2010. [24](#)
- [Qi 93] X. Qi and F. Palmieri. “The Diversification role of crossover in the genetic algorithm”. In : *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 132 – 137, Urbana, Illinois , USA, July 17 - 21 1993. [37](#)
- [Ray 53] T. Ray and K. Liew. “A swarm metaphor for multiobjective design optimization”. *Engineering Optimization*, Vol. 34, No. 2, p. 2002, 141-153. [29](#)
- [Reye 06] M. Reyes-Sierra and A. C. Coello. “Multi-Objective Particle Swarm Optimizers : A survey of the state-of-the-art”. *Computational Intelligence Research*, Vol. 2, No. 3, pp. 287–308, 2006. [30](#)
- [Rich 89] J. T. Richardson. “Some guidelines for genetic algorithms with penalty functions”. In : *Proceeding of the 3th International Conference on Genetic Algorithms*, pp. 191 – 197, Fairfax, Virginia, USA, June 4 - 7 1989. [28](#)
- [Roch 10] R. Roche, B. Blunier, and A. Miraoui. “Multi-Agent Systems For Grid Energy Management : A Short Review”. In : *Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3341 – 3346, Phoenix, Arizona, USA, November 7 - 10 2010. [59](#)
- [Roch 11a] R. Roche, L. Idoumghar, B. Blunier, and A. Miraoui. “Imperialist Competitive Algorithm for Dynamic Optimization of Economic Dispatch in Power Systems”. In : *Proceedings of Interssnational Conference on Artificial Evolution*, pp. 1–12, Angers, France, October 24-26 2011. [57](#), [71](#), [107](#)
- [Roch 11b] R. Roche, L. Idoumghar, B. Blunier, and A. Miraoui. “Optimized Fuel Cell Array Energy Management Using Multi-Agent Systems”. In : *Proceedings of 46th Annual Conference of IEEE Industry Applications Society*, pp. 1–12, Orlando, FL, USA, October 9-13 2011. [57](#), [71](#)
- [Roch 11c] R. Roche, L. Idoumghar, A. Miraoui, B. Blunier, M.Daggag, C. Solacolu, and A. Boyett. “Système et procédé de commande d'une installation de production d'énergie électrique”. Tech. Rep. B11-0627FR, 2011. General Electric. [57](#)
- [Roch 12] R. Roche, L. Idoumghar, S. Suryanarayanan, M. Daggag, C.-A. Solacolu, and A. Miraoui. “A Flexible and Efficient Multi-Agent Gas Power Plant Energy Management System With Economic and Environmental Constraints”. *Applied Energy*, June 2012. Accepted for publication. [57](#), [107](#)

Références bibliographiques

- [Scha 85] J. D. Schaffer. "Multiple objective optimization with vector evaluated genetic algorithm". In : *Proceedings of the first International Conference on Genetic Algorithms*, pp. 93 – 100, Pittsburgh, PA, USA, July 24 - 26 1985. [28](#)
- [Sera 94] P. Serafini. "Simulated annealing for multiple objective optimization problems". In : *Proceedings of the 10th International Conference on Multiple Criteria Decision Making : Expand and Enrich the Domains of Thinking and Application*, pp. 283–292, Springer, Berlin, deutschland, August 1994. [27](#)
- [Shar 10a] D. Sharma and P. Collet. "An archived-based stochastic ranking evolutionary algorithm (ASREA) for multi-objective optimization". In : *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pp. 479–486, Portland, Oregon, USA, July 7-11 2010. [11](#)
- [Shar 10b] D. Sharma and P. Collet. "GPGPU-compatible archive based stochastic ranking evolutionary algorithm (G-ASREA) for multi-objective optimization". In : *Proceedings of the 11th international conference on Parallel problem solving from nature : Part II*, pp. 111–120, Springer-Verlag, Kraków, Poland, September 11-15 2010. [12](#)
- [Shi 98] Y. Shi and R. C. Eberhart. "A Modified Particle Swarm Optimizer". In : *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 69–73, IEEE, Anchorage, Alaska, May 4 - 9 1998. [23](#)
- [Shi 99] Y. Shi and R. C. Eberhart. "Empirical Study of Particle Swarm Optimization". In : *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1945–1950, IEEE, Washington, DC, USA, July 10 - 16 1999. [22](#)
- [Siar 07] P. Siarry. *Optimisation en traitement du signal et de l'image*. Hermès, 2007. [15](#)
- [Siar 94] P. Siarry. *La méthode du recuit simulé en électronique. Adaptation et accélération. Comparaison avec d'autres méthodes d'optimisation. Applications dans d'autres domaines*. PhD thesis, 1994. [14](#)
- [Simo 11] M. Simoes, R. Roche, E. Kyriakides, A. Miraoui, B. Blunier, K. McBee, S. Suryanarayanan, P. Nguyen, and P. Ribeiro. "Smart-Grid Technologies and Progress in Europe and the USA". In : *Proceedings of the IEEE Energy Conversion Congress & Exposition (ECCE 2011)*, pp. 383 – 390, Phoenix, Arizona, USA, September 17 - 22 2011. [58](#)
- [Sing 97] S. Singh and D. Bertsekas. "Reinforcement learning for dynamic channel allocation in cellular telephone systems". In : *Proceedings of the 1996 Conference in Advances in Neural Information Processing Systems*, pp. 974 – 980, Denver, CO, USA, December 2 - 5 1997. [32](#)
- [Siva 07] S. N. Sivanandam and S. N. Deepa. *Introduction to Genetic Algorithms*. Springer Publishing Company, Incorporated, 2007. [16](#)

- [Srin 03] D. Srinivasan and T. Seow. “Particle swarm inspired evolutionary algorithm for multiobjective optimization problem”. In : *Proceedings of the 2003 Congress on Evolutionary Computation*, pp. 2292 – 2297, Canberra, Australia, December 8 - 12 2003. [30](#)
- [Sriv 95] N. Srivinas and K. Deb. “Multiobjective Optimization using Non-Dominated Sorting in Genetic Algorithms”. *Evolutionary Computation*, Vol. 2, No. 8, pp. 221–248, 1995. [29](#)
- [Stei 02] S. Steinke, L. Wehmeyer, B. Lee, and P. Marwedel. “Assigning Program and Data Objects to Scratchpad for Energy Reduction”. In : *Proceedings of the conference on Design, automation and test in Europe*, pp. 409 – 415, IEEE, Munich, Germany, March 12 - 16 2002. [85](#)
- [Stut 99] T. Stutzle and H. Hoos. *MAX-MIN ant system and local search for combinatorial optimization problems*. Kluwer, Boston, meta-heuristics : advances and trends in local search paradigms for optimization Ed., 1999. [21](#)
- [Suga 05] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization”. Technical Report 2005005, Nanyang Technological University, Singapore and IIT Kanpur, India, May 2005. [108](#)
- [Sydo 09] M. Sydow, F. Bonchi, C. Castillo, and D. Donato. “Optimising Topical Query Decomposition”. In : *Proceedings of the 2009 workshop on Web Search Click Data*, pp. 43 – 47, ACM, Barcelona, Spain, February 9 2009. [15](#)
- [Sysw 89] G. Syswerda. “Uniform crossover in genetic algorithms”. In : *Proceedings of the 3th International Conference on Genetic Algorithms*, pp. 2 – 9, Fairfax, Virginia, USA, June 4 - 7 1989. [37](#)
- [Talb 02] E. Talbi. “A taxonomy of Hybrid Metaheuristics”. *Journal of Heuristics*, Vol. 8, No. 5, pp. 541–564, 2002. [12](#), [23](#)
- [Talb 99] E. Talbi. “Métaheuristiques pour l’optimisation combinatoire multi-objectif : Etat de l’art”. Technical Report PE : 98-757.33, CNET, 1999. [26](#), [28](#), [29](#)
- [Tane 05] A. Tanenbaum. *Architecture de l’ordinateur. 5e édition*, November 2005. [79](#)
- [Tiou 95] S. Tiourine, C. Hurkens, and J. Lenstra. “An overview of algorithmic approaches to frequency assignment problems”. In : *Proceedings of the Calma Symposium on Combinatorial Algorithms for Military Applications*, pp. 53–62, Scheveningen, netherlands, November 24 1995. [32](#)
- [Tiwa 09] S. Tiwari, G. Fadel, P. Koch, and K. Deb. “Performance assessment of the hybrid Archive-based Micro Genetic Algorithm (AMGA) on the CEC09 test problems”.

Références bibliographiques

- In : *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1935 – 1942, IEEE, Trondheim, Norway, May 18-21 2009. [11](#), [104](#)
- [Truo 98] D. N. Truong, F. Bodin, and A. Seznec. “Improving Cache Behavior of Dynamically Allocated Data Structures”. In : *Proceedings of the 1998 International Conference on Parallel Architectures and Compilation Techniques*, pp. 322 – 329, Paris, France, October 12 - 18 1998. [85](#)
- [Uday 02] S. Udayakumaran. *Application Specific Memory Partitioning for Low Power*. George Washington University, 2002. [85](#)
- [Ulun 94] E. Ulungu and J. Teghem. “Multi-objective combinatorial optimization : a survey”. *Multi-Criteria Decision Analysis*, Vol. 3, pp. 83–104, 1994. [26](#)
- [Ulun 99] E. L. Ulungu, J. Teghem, P. Fortemps, and D. Tuyttens. “MOSA method : A tool for solving multi-objective combinatorial optimization problems”. *Multicriteria Decision Analysis*, Vol. 20, pp. 221–236, 1999. [27](#)
- [Vasq 01] M. Vasquez and J. Hao. “A Heuristic Approach for Antenna Positioning in Cellular Networks”. *Journal of Heuristics*, Vol. 7, pp. 443 – 472, 2001. [32](#)
- [VHF 95] *VHF and UHF propagation curves for frequency range from 30 Mhz to 1000 Mh*. 1995. [44](#)
- [Vill 11] J. Villegas. “Using nonparametric test to compare the performance of metaheuristics”. 2011. <http://juangvillegas.files.wordpress.com/2011/08/friedman-test-24062011.pdf>. [119](#)
- [Wang 04] X. Wang and J. Li. “Hybrid Particle Swarm Optimization with Simulated Annealing”. In : *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, pp. 2402–2405, Shanghai, China, August 26 - 29 2004. [24](#)
- [Wang 07] Y. Wang, Z. Zhao, and R. Ren. “Hybrid Particle Swarm Optimizer with Tabu Strategy for Global Numerical Optimization”. In : *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2310–2316, Singapore, September 25 - 28 2007. [24](#)
- [Wehm 04] L. Wehmeyer, U. Helmig, and P. Marwedel. “Compiler-Optimized Usage of Partitioned Memories”. In : *Proceedings of the 3rd workshop on Memory performance issues : in conjunction with the 31st international symposium on computer architecture*, pp. 114–120, ACM, Munich, Germany, June 20 2004. [85](#)
- [Xia 06] W. Xia and Z. Wu. “A Hybrid Particle Swarm Optimization Approach for the Jobshop Scheduling Problem”. *Advanced Manufacturing Technology*, Vol. 29, No. 3-4, pp. 360–366, 2006. [88](#)

- [Yu X 10] W. Yu-Xuan, X. Qiao-Liang, and Z. Zhen-Dong. “Particle Swarm Optimizer with Adaptive Tabu and Mutation : A Unified Framework for Efficient Mutation Operators”. *ACM Transactions on Autonomous and Adaptive Systems*, Vol. 5, pp. 1 :1 – 1 :27, February 2010. [25](#), [122](#), [124](#)
- [Zend 06] O. Zendra. “Memory and Compiler Optimizations for Low-Power and Energy”. In : *Proceedings of the International Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems*, p. 8, Nantes, France, July 3 2006. [80](#)
- [Zhan 89] M. Zhang and T. Yum. “Comparisons of channel assignment strategies in cellular mobile telephone systems”. *IEEE Transaction on Vehicular Technology*, Vol. 38, No. 4, pp. 211 – 215, 1989. [32](#)
- [Zhen 99] Y. Zheng and S. Kiyooka. “Genetic Algorithm Applications”. 1999. www.me.uvic.ca/~zdong/courses/mech620/GA_App.PDF. [18](#)
- [Zitzl 01] E. Zitzler, M. Laumanns, and L. Thiele. “SPEA2 : Improving the Strength Pareto Evolutionary Algorithm for multiobjective optimization”. In : *Proceeding of the Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pp. 95–100, Athens, Greece, September 19-21 2001. [29](#)
- [Zitzl 99] E. Zitzler and L. Thiele. “Multiobjective Evolutionary Algorithms : A Comparative Case Study and the Strength Pareto Approach”. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257–271, 1999. [29](#), [30](#)

Références bibliographiques

Annexe : Quelques unes de mes publications

Mes publications jointes à ce mémoire sont :

- **L. Idoumghar** and R. Schott, “Two Distributed Algorithms for the Frequency Assignment Problem in the Field of RadioBroadcasting”. Journal of IEEE Transactions on Broadcasting, Vol. 55, Issue 2, Part 1, pp. 223-229, June 2009.
- **L. Idoumghar** and Ph. Debreux, “New Modeling Approach for the Frequency Assignment Problem in Broadcasting”. Journal of IEEE Transactions on Broadcasting, pp. 293-298, Vol. 48, N. 4, December 2002.
- R. Roche, **L. Idoumghar**, B. Blunier and A. Miraoui, “Optimized Fuel Cell Array Energy Management Using Multi-Agent Systems”, 46th Annual Conference of IEEE Industry Applications Society (IAS 2011), October 9-13, 2011, Orlando, FL, USA (CD-ROM).
- R. Roche, **L. Idoumghar**, B. Blunier and A. Miraoui, “Imperialist Competitive Algorithm for Dynamic Optimization of Economic Dispatch in Power Systems”, International Conference on Artificial Evolution (EA-2011), October 24-26, 2011, Angers, (CD-ROM).
- **L. Idoumghar**, M. Idrissi-Aouad, M. Melkemi and R. Schott, “Hybrid PSO-SA Type Algorithms for Multi-Modal Function Optimization and Reducing Energy Consumption in Embedded Systems”. Journal of Applied Computational Intelligence and Soft Computing, vol. 2011, pp. 1-12 , juin 2011.
- **L. Idoumghar**, M. Idrissi-Aouad, M. Melkemi and R. Schott, “Metropolis Particle Swarm Optimization Algorithm with Mutation Operator For Global Optimization Problems”. 22th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2010), pp. 35-42, October 27-29, 2010, Arras, France.
- **L. Idoumghar**, D. Fodorean, A. N’diaye, D. Bouquain and A. Miraoui, “Simulated Annealing Algorithm for the Optimization of an Electrical Machine”. Journal of IET Electric Power Applications (accepted for publication).
- **L. Idoumghar**, M. Idrissi-Aouad, M. Melkemi and R. Schott, “Hybrid PSO-SA Type

Annexe : Quelques unes de mes publications

Algorithms for Multi-Modal Function Optimization and Reducing Energy Consumption in Embedded Systems”. Journal of Applied Computational Intelligence and Soft Computing, vol. 2011, pp. 1-12 , juin 2011.

- **L. Idoumghar**, M.Idrissi-Aouad, M. Melkemi and R. Schott, “Metropolis Particle Swarm Optimization Algorithm with Mutation Operator For Global Optimization Problems”. 22th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2010), pp. 35-42, October 27-29, 2010, Arras, France.

New Modeling Approach to the Frequency Assignment Problem in Broadcasting

Lhassane Idoumghar and Philippe Debreux

Abstract

This paper presents an original algorithm that uses a new modeling approach of the interference constraints and uses a *probabilistic taboo search* algorithm to solve the frequency assignment problem in the field of broadcasting. The results obtained by our algorithm are compared to the operating solutions in the field of FM broadcasting in France and the best known results obtained by hybrid genetic algorithm [2] that uses a classical modeling of the interferences constraints. By analysing the results obtained by our approach we can observe that we efficiently enhance the quality of the solutions.¹

Index Terms

Taboo Search, Frequency Assignment Problem, Radiobroadcasting.

I. INTRODUCTION

IN order to perform suitable geographic and/or demographic coverages for broadcasting or radiocommunications, it is necessary to distribute a large number of transmitters over the geographic area. To each transmitter is assigned one frequency. Unfortunately, only a finite frequency spectrum is available for a given system, which makes it necessary to reuse frequencies. This problem generates interference phenomena due to co or adjacent frequencies allocated to neighbouring transmitters. Here, it is of major importance to minimize the interference while at the same time use the spectrum efficiently. This problem, known to be NP-Hard², can be stated as finding an optimal assignment of frequencies to a set of transmitters, under various constraints. This problem can be represented as a generalised graph colouring problem. Transmitters are represented by the vertices of a constraints graph [4]. In the classical approach of the frequency assignment problem, each edge of the graph represents a constraint (one value) between the frequencies assigned to the two transmitters connected by this edge.

The graph colouring problem is one of the most studied NP-Hard problems, and can be defined informally as follows. Given an undirected graph, one wishes to colour, with a minimal set of colours, the vertices of the graph in such a way that two adjacent vertices are assigned different colours. Due to the NP-Hardness of this problem [5] heuristic methods must be used for large graphs. Many algorithms have been developed: Branch and Bound optimisation [6], Lagrange optimisation [7], constructive methods, local search meta-heuristics [8], [9], evolutionary algorithms [10], hybrid algorithms [2], [11], [12], ant colony paradigms [13], [14], constraints programming [15], simulated annealing [16], artificial neural networks [17], [18] and Markov Decision Processes [19].

The original algorithm presented in this paper uses, conversely to the classical approach, a new modeling of the interferences constraints. To find a best frequency assignment our algorithm uses a new *probabilistic taboo search*. An application of this new algorithm in the field of broadcasting leads to an enhancement of the quality of the solutions compared to several other approaches like hybrid genetic algorithms.

This paper is organized as follows: Section 2 recalls the classical approach of the frequency assignment problem. Section 3 gives a review of taboo search. Section 4 describes our approach. In Section 5 we present some experimental results. Concluding remarks and further research aspects are contained in the last section.

L. Idoumghar and Ph. Debreux are with the TDF-C2R Broadcasting and Wireless Research Center, 1 rue Marconi, Technopôle Metz 2000, 57078 Metz, France. E-mail: {lhassane.idoumghar, philippe.debreux}@tdf.fr. Tel: +33 3 87 20 75 27. FAX: +33 3 87 74 54 30.

¹This work is currently under the ongoing process of patent pending. A preliminary version of this paper will be presented at SEA'2002 [1].

²The frequency assignment problem may be reduced to the graph colouring problem which is NP-Complete [3] combinatorial problem. This means that the time needed to compute an optimal solution increases exponentially with the size of the problem.

II. FREQUENCY ASSIGNMENT PROBLEM

The frequency assignment problem arises when a large number of transmitters are operating in a region, and the interference caused by transmitters on receivers served by other transmitters has to be minimized. This can be normally achieved easily if the number of available frequencies is large enough. The problem is to minimize the interference while at the same time using the spectrum efficiently.

As presented in the first section, the frequency assignment problem can be represented in a graph theoretic formulation. The transmitters are represented by vertices V of a constraint graph G . In the classical approach of the frequency assignment problem, the graph $G=(V,E)$ is defined as an undirected graph, where $V = \{v_1, \dots, v_n\}$ is the set of vertices and $E = \{(v_i, v_j) / v_i, v_j \in V\}$ is the set of edges. This graph is built on an adjacency symmetric matrix $M = (m_{i,j})^{n \times n}$ (called constraints matrix) defined by:

- n represents the number of vertices of the graph,
- $m_{i,j}$ with $i \neq j$ represents the minimum frequency separation required to satisfy the constraint between vertices v_i and v_j ($m_{i,j}$ is a non-negative integer),
- $m_{i,j} = 0$ means that there is no constraint between vertices v_i and v_j ,
- $m_{i,i} = 0$ means that an emitter has no constraint with itself.

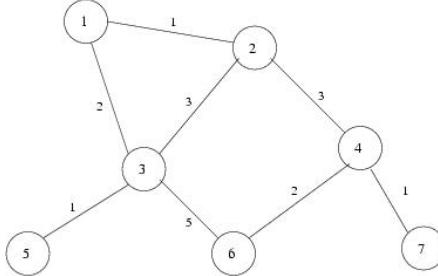


Fig. 1. Example of the constraints graph.

Figure 1 gives an example of the constraints graph that corresponds to the following constraints matrix:

$$M = \begin{pmatrix} 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 3 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 1 & 5 & 0 \\ 0 & 3 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Informally, a frequency assignment problem is a mapping $f : V \rightarrow C$ (where C is a set of integers $0, \dots, k$ representing frequencies) such that the constraints $|f(v_i) - f(v_j)| > m_{i,j}$ are satisfied for all $v_i, v_j \in E$. A feasible solution of this problem is represented by a vector $\vec{C}(c_0, c_1, \dots, c_k)$ where $c_i \in C$ for all $i \in [0, k]$. The components of \vec{C} are essentially the frequencies assigned to the vertices of G .

To generate a constraints matrix M some propagation models are used [20]. These models use approximations to evaluate the constraint $m_{i,j}$ required to satisfy the interference between the two transmitters v_i and v_j . Often, these approximations represent a disadvantage of the classical modeling because it supposes that all these constraints are representatives. Some algorithms [2], [13], [21] that are used to solve the frequency assignment in the field of broadcasting use this modeling. Even if these algorithms obtained the best solutions, that minimized the number of violated constraints, an analysis of these results under real propagation conditions shows that quality of these solutions is not better than the operating solution in the field of broadcasting.

To avoid the drawback of the classical modeling, we have developed a new modeling approach that is presented in section 4.

III. TABOO SEARCH

The basic idea of the taboo (or *tabu*) search metaheuristic [8], [22], [23] is to explore the *search space* of all feasible solutions by a sequence of moves. A move from one solution to another is generally the best available. However, in order to prevent oscillation and to provide a mechanism for escaping from locally optimal but not globally optimal solutions, some moves, at one particular iteration, are classified as forbidden or *taboo*. Moves are regarded as taboo by consideration of short-term and long-term history of the sequence of moves. A very simple use of this idea might be to classify a move as taboo if the reverse move has been made recently or frequently. There is also an *aspiration criterion*, which overrides the taboo moves in particular circumstances. These circumstances might include the cases when, by forgetting that a move is taboo, a solution which is the best so far is obtained.

Suppose it is required to minimize some cost function F on the search space S . For combinatorially hard problems, such as the frequency assignment problem, it may only be possible to obtain sub-optimal solutions, in which F is close to its minimum value. Sub-optimal problems may be obtained when a certain threshold for an acceptable solution has been achieved or when a certain number of iterations have been completed.

A characterisation of the search space S for which taboo search can be applied is when there exists a set of k moves $D = \{d_1, \dots, d_k\}$ such that the application of the moves to a feasible solution $s \in S$ leads to k , usually distinct, solutions $D(s) = \{d_1(s), \dots, d_k(s)\}$. The subset $N_{set}(s) \subseteq D(s)$ of feasible solutions is known as the neighbourhood of s .

The method starts with (possibly random) solution $s_0 \in S$ and determines a sequence of solutions $s_0, s_1, \dots, s_n \in S$. At each iteration, s_{j+1} ($0 \leq j < n$) is selected from the neighbourhood $N_{set}(s_j)$. The selection process is first to determine the taboo set $T_{set}(s_j) \subseteq N_{set}(s_j)$ and the aspiration set $A_{set}(s_j) \subseteq N_{set}(s_j)$. Then s_{j+1} is the neighbour of s_j which is either an aspirant or not taboo and for which $F(s_{j+1})$ is minimal; that is, $F(s_{j+1}) \leq F(s_i)$ for all $s_i \in (N_{set}(s_j) - T_{set}(s_j)) \cup A_{set}(s_j)$.

In the next section we will present different elements used by our algorithm based on a refined modeling of the input data and on a *probabilistic taboo search*.

IV. OUR APPROACH

A. Directed Constraints Graph

The first step of the planning process, called *dimensionning*, consists in spreading an optimal number of transmitters over an area in order for this area to be efficiently covered. This number should be as small as possible for obvious economic reasons. Transmitters are spread over the geographical area where providers wish to provide the users with their services, each transmitter covers a part of this geographical area called its *coverage area* (Figure 2). The area around a transmitter where transmission conditions are favourable enough to have a good reception of the signal is known as the *service area* of the transmitter. The service area is the portion of the coverage area that is not subject to interference by other transmitters.

The transmission quality and thus the shapes of these regions will depend heavily on the propagation conditions and current interferences produced by the other transmitters. Therefore, coverage and service areas are frequently of highly irregular shape.

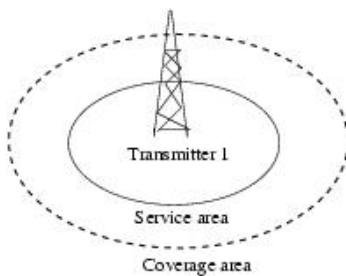


Fig. 2. Coverage and service area of a transmitter

Let V be a set of transmitters distributed across the geographical area. These transmitters represent the vertices of the constraints graph. We recall that in the FM radiobroadcasting domain, the constraint $C_{i,j} \in [1, 5]$ that might exist between two transmitters v_i and v_j is defined as $|f_i - f_j| < C_{i,j}$.

To build our constraints graph we must compute for each pair $\langle v_i, v_j \rangle$ (resp. $\langle v_j, v_i \rangle$) the list $L_{i,j}$ (resp. $L_{j,i}$) where $i, j \in V$ and $|L| = 5$. For each list $L_{i,j}$, the index corresponds to a minimal gap in frequency that is required to avoid interference phenomena between the two transmitters v_i and v_j . The value $L_{i,j}(k)$, where $k \in [0, 4]$, represents a number of points, of the coverage area of transmitter v_j , that are subject to interference by transmitter v_i . There exists an edge between a pair of transmitter $\langle v_i, v_j \rangle$ if and only if $L_{i,j}(0) \neq 0$. This means that there exist subject to interference points when the same frequency is assigned to the two transmitters v_i and v_j . An example of a constraints graph is given in Figure 3. In this example, we can see that the transmitter v_j interferes with the transmitter v_k (there exists a list $L_{j,k}$). But, the transmitter v_k does not interfere with the transmitter v_j because there is no edge between the corresponding vertices ($L_{k,j}(0) = 0$).

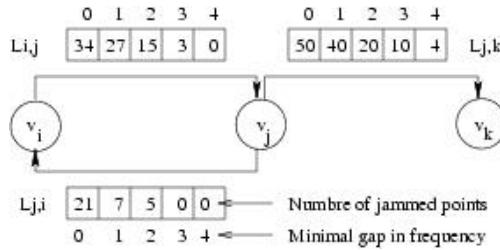


Fig. 3. Example of a constraints graph.

The second step of the planning process, called *frequency assignment*, looks for a best frequency to be assigned to each transmitter. Due to the NP-Hardness of this problem we have developed an original probabilistic taboo search described in the next subsection.

B. Probabilistic taboo search

In this subsection we describe how the probabilistic taboo search procedure has been implemented to the frequency assignment problem. We explain the representation of assignments, the notions of neighbourhood and the cost function that we used to evaluate all neighbours of an assignment.

Representation of an assignment and neighbourhood

A frequency assignment $X(f_1, f_2, \dots, f_n)$ is represented using an array of indexes $[x_1, \dots, x_n]$ where $X_i = a_{x_i}$ for $1 \leq i \leq n$ (f_i represents the frequency assigned to the transmitter e_i). The neighbours N of X are those assignment where the array of indexes differs in precisely one component. Thus X' is represented by $[x'_1, \dots, x'_n]$ then $X'(f'_1, f'_2, \dots, f'_n)$ is a neighbour of X if there exists j , $1 \leq j \leq n$, such that $x_j \neq x'_j$, and for all $i = 1, \dots, n$ with $i \neq j$ we have $x_i = x'_i$.

Cost function

In order to guide the probabilistic taboo algorithm to visit the search space, we need a cost function that will serve to evaluate the solutions. In this work, for a given solution $X(f_1, f_2, \dots, f_n)$, $|X| = n$, a solution is evaluated by using the following cost function:

$$F(E) = \sum_{j=1}^n (\phi(e_j, f_j) + \psi(e_j, f_j)).weight(e_j)$$

where:

- f_j represents the frequency assigned to the transmitter e_j ,
- $\phi(e_j, f_j)$ gives a number of points, of the coverage area of the transmitter e_j , that are subject to interference by other transmitters when f_j is assigned to transmitter e_j ,
- $\psi(e_j, f_j)$ gives a number of points, of the coverage area of the other transmitters, that are subject to interference by the transmitter e_j when f_j is assigned to transmitter e_j ,

- $weight(e_j)$ represents the weight associated to the transmitter e_j . This weight depends to the power of e_j . This parameter serves to give more importance to the transmitters that have a low power.

The aim of the cost function is to minimize the number of subject to interference points.

Taboo list

When the frequency of transmitter e_i in a configuration X is changed to a new value, the value $\langle e_i \rangle$ is classified taboo for l iterations (*taboo tenure*). That is, the new frequency will not be changed to e_i during this period.

Algorithm

The algorithm used to find a best frequencies assignment is summarized in figure 4.

As we say in section 3, at each iteration, taboo search selects one of the best neighbouring configurations among the ones that are not forbidden by the taboo list, even if the chosen configuration is not better than the current one. It is well known that at each iteration, taboo search chooses a neighbour s_{j+1} of the current configuration s_j from the neighbourhood set $N_{set}(s_j)$. The selection process consists in (1) determining the taboo set $T_{set}(s_j) \subseteq N_{set}(s_j)$ and (2) determining the aspiration set $A_{set}(s_j) \subseteq N_{set}(s_j)$. In our taboo search we introduce a probability p_t to choose one configuration from $A_{set}(s_j)$ and we use a *uniform probability* to choose one configuration from $N_{set}(s_j) - T_{set}(s_j)$. An implementation of probabilistic taboo search in the domain of radiobroadcasting shows that a better result is obtained with a low value of probability p_t .

```

construct the constraint graph  $G$  (Directed graph)
generate an initial solution  $x^*$ 
// let  $x^*$  be a best solution
For  $i := 1$  to  $nbOpt$  Do
    stop  $\leftarrow$  false
    taboo_list  $\leftarrow \emptyset$ 
    let  $x \leftarrow x^*$  be a current solution  $x(j)$ , with  $j \in [1, n]$ 
        and  $|x| = n$ , represents frequency assigned to transmitter j
    While not stop Do
        with a probability  $p_t$ , choose  $s_t$  in the best neighbouring
        configuration forbidden by taboo list
        let  $s^*$  be the best neighbouring configuration which is not
            forbidden by the taboo list. This configuration
            is chosen by using a uniform probability
        choose the best configuration  $y$  between  $s_t$ 
            and  $s^*$  which minimizes a cost function
        newf  $\leftarrow$  search to  $y$  a best frequency
            that minimizes a cost function
         $x(y) \leftarrow newf$ 
        adjust taboo_list
        If  $x$  is better than  $x^*$  Then
             $x^* \leftarrow x$ 
            stop  $\leftarrow$  stopping criterion
    Done
EndFor
```

Fig. 4. Probabilistic taboo search.

In figure 4, $nbOpt$ specify the number of optimizations that the algorithm can do. After the first optimization the algorithm starts with the last best solution that is found. The number of iterations necessary at each optimization step $j \in [2, nbOpt]$, is relatively small because the search process starts from the last best solution and just tries to explore the neighbouring, initializing the taboo list at \emptyset . The *stop* criterion is used to break the search process when this process is not able to improve the solution.

V. EXPERIMENTS

This section presents the main results obtained after several runs performed over two real instances of frequency assignment problems provided by TDF-C2R Broadcasting and Wireless Research Center. These instances have a very big size in terms of number of vertices (transmitters) and edges (interference constraints). Each vertex may take up $n = 200$ possible values (bandwidth 87.5 - 107.5 MHz, $|107.5 - 87.5 \text{ MHz}| = 20 \text{ MHz} = n * 100 \text{ kHz}$). We recall that the main objective is to minimize a number of subject to interference points and to compare a refinement of our modeling approach with the classical approach used to generate interferences constraintes.

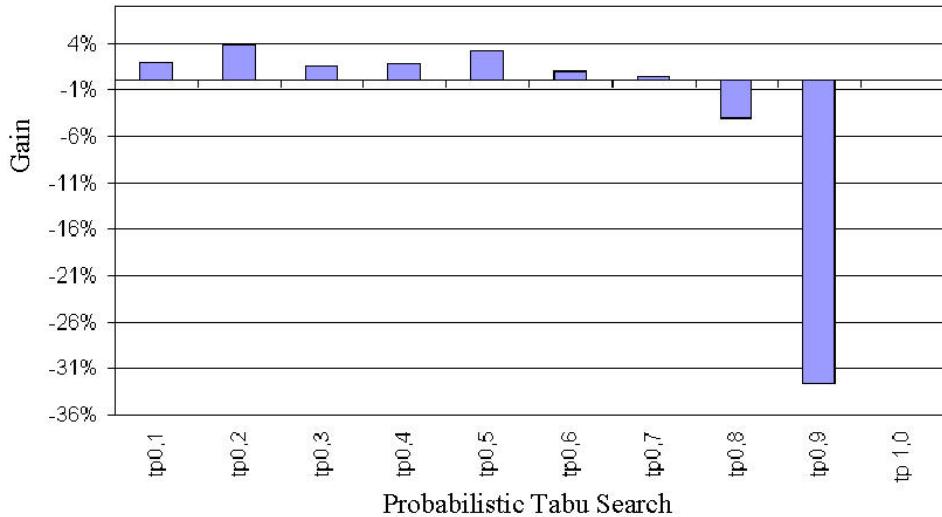


Fig. 5. Different gains obtained by using probabilistic tabu search compared to the best results given in [21].

A. Aspiration criterion

In order to evaluate the probability p_t , that is used in the aspiration criterion, we have used a probabilistic taboo search and a classical modeling of the interferences constraints to resolve one real instance (more than 5790 transmitters and more than 153484 interferences constraints) of the frequency assignment problem in the field of FM radiobroadcasting in France.

The results obtained are presented in figure 5. The gain represented in this figure corresponds to a percentage of violated constraints that we satisfy/violate, when we use different probabilities, compared to the best results obtained by [21].

In Figure 5, tpx means that a probabilistic tabu search uses an aspiration criterion with probability $p_t = x$. As we can see, a best gain is obtained with a probability $p_t = 0.2$.

B. Quality plan estimation in regard of radio system characteristics

In this section we present a validation of our results under real propagation conditions. We calculate for each transmitter a quality index which can be defined as a *service area /coverage area* ratio. We recall that the coverage area of each transmitter is defined as the area around this transmitter where transmission conditions are favourable enough to have a good reception of the signal. The service area is the portion of the coverage area that is not subject to interference by other transmitters. The example given in figure 6 shows how the service area of *transmitter1* is calculated. In this figure, the points of the coverage area of *transmitter1* that are not subject to interference by other transmitters form the service area of *transmitter1*.

The results given in figure 7 and in figure 8 correspond respectively to the results obtained by solving two instances of a real radio networks:

- *instance1*: more than 5790 transmitters and more than 3739 edges,

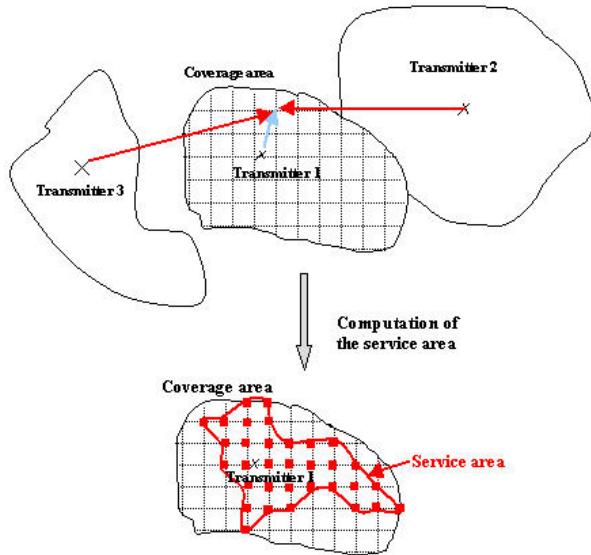
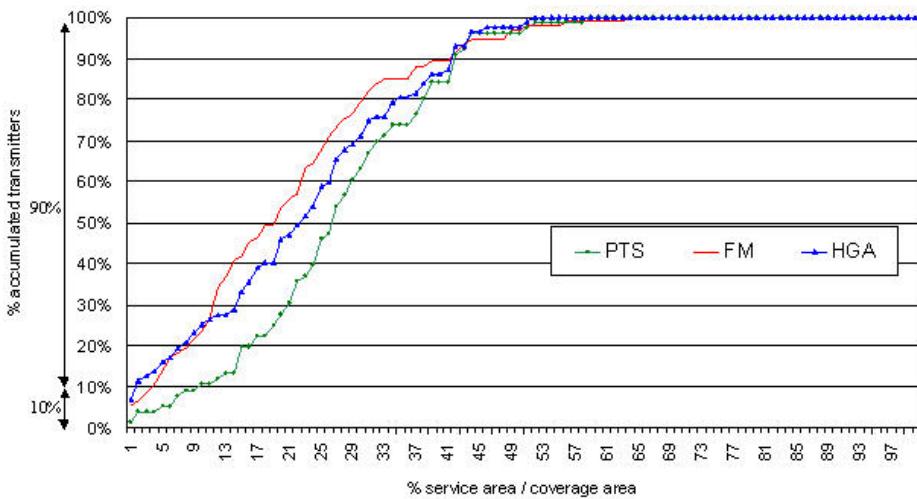


Fig. 6. Computation of the service area

- *instance2*: more than 5790 transmitters and more than 63692 edges.

In these figures, the curve called *HGA* corresponds to the best solution obtained by using hybrid genetic algorithm [2] that uses a classical approach to generate interferences contraintes. The curve called *FM* corresponds to the operating solution in the field radiobroadcasting in France and the curve called *PTS* corresponds to the best solution that is obtained by the approach described in this paper.

Fig. 7. Comparison of three best solutions obtained by solving *instance1*

As we can observe in figure 7:

- on the *FM* curve, 90% of the transmitters have a service area at least equal to approximatively 3% of their coverage area,
- on the *HGA* curve, 90% of transmitters have a service area at least equal to approximatively 1% of their coverage area,

- on the *PTS* curve, 90% of transmitters have a service area at least equal to approximatively 12% of their coverage area,

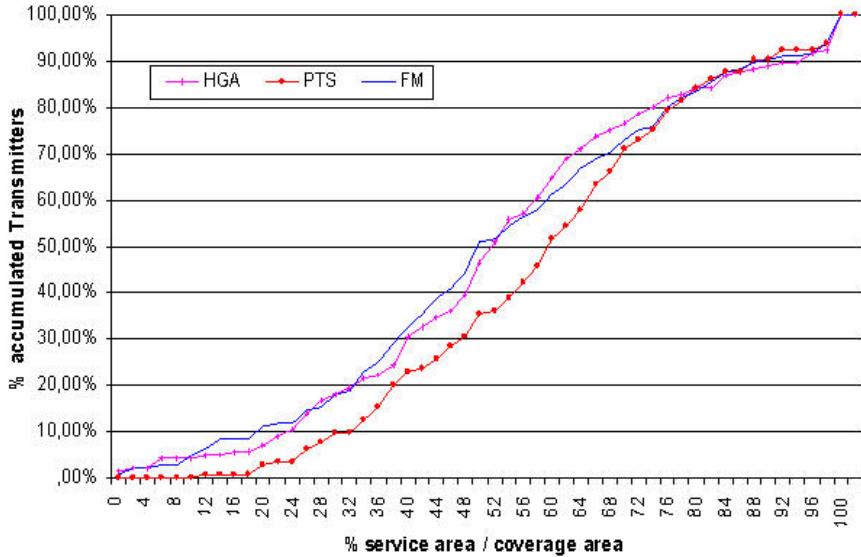


Fig. 8. Comparison of three best solutions obtained by solving *instance2*

By doing the same comparison of the curves that are given in figure 8, we can see that:

- on the *FM* curve, 90% of the transmitters have a service area at least equal to approximatively 18% of their coverage area,
- on the *HGA* curve, 90% of transmitters have a service area at least equal to approximatively 22% of their coverage area,
- on the *PTS* curve, 90% of transmitters have a service area at least equal to approximatively 32% of their coverage area,

The curves given in figure 7 and 8, show that the results obtained by using our approach are better than the results obtained by using a hybrid genetic algorithm [2] and the operating solutions in the field of radiobroadcasting in France.

VI. CONCLUDING REMARKS AND FURTHER RESEARCH ASPECTS

In this paper we have presented a new modeling approach that is used to solve the frequency assignment problem. This approach uses a new modeling of the constraints graph and uses an original probabilistic taboo search to find a best frequency to be assigned to each transmitter. The experiments on real radio networks show that the quality of the results obtained by using the approach presented in this paper can be improved compared to the use of hybrid genetic algorithm [2]. Also a comparison of our solutions with the operating solutions in the field of radio broadcasting shows that our modeling approach gives good solutions (in terms of quality of the solutions).

As an aspect of future work, extended experimentation on a variety of radio networks is expected. Furthermore, we are currently working to implement this algorithm on parallel machines.

Acknowledgments

The authors are grateful to anonymous referees for their pertinent comments and suggestions. Also, they wish to thank Mr. Thierry Schott, research engineer at TDF-C2R, for several useful suggestions.

REFERENCES

- [1] L. Idoumghar and Ph. Debreux, New Modeling Approach to the Frequency Assignment Problem in Broadcasting, *Proc. of SEA'2002, IASTED International Conference on Software Engineering and Applications*, pp. 807-812, November 4-6, 2002, Cambridge, USA .
- [2] L. Idoumghar and M. Alabau and R. Schott, New Hybrid Genetic Algorithms for the Frequency Assignment Problem, *Journal of IEEE Transactions on Broadcasting*, March 2002, Vol. 48, N 1, 27-34.
- [3] A. Gamst and W. Rave, on the Frequency Assignment in mobile automatic telephone system, *Proc of GLOBECOM*, pp. 305-315, 1982.
- [4] S. Hurley and D.H. Smith and S.U. Thiel, Fasoft: A System for Discrete Frequency Assignment, *Radio Science*, 32(5), 1997, 1921-1939.
- [5] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness* (W.H. Freeman and Co., 1979).
- [6] H.W. Fastert, *The Mathematical Theory Underlying the Planning of Transmitter Networks*, EBU Review, pp. 60-69, April 1960.
- [7] H. Everett, Generalized Lagrange Multiplier Method for Solving Problems of Optimal Allocation of Resources, *Operations Research*, Vol. 11, pp. 397-417, 1963.
- [8] F. Glover and E. Taillard, A User's Guide to Tabu Search, *Annals of Operations Research*, 41, 1993, 3-28.
- [9] J. K. Hao, R. Dore and Ph. Galinier, Tabu Search for Frequency Assignment in Mobile Radio Networks. *Journal of Heuristics*, 4, 1998, 47-62.
- [10] J.K. Hao and R. Dorne, Study of Genetic Search for the Frequency Assignment Problem, *Artificial Evolution*, Lecture Notes in Computer Science 1063, SpringerVerlag 1996, 333-344.
- [11] C. Fleurent and J.A. Ferland, Genetic and Hybrid Algorithms for Graph Coloring, *Annals of Operations Research*, 63, 1995, 437-463.
- [12] R. Dorne and J. K. Hao, A New Genetic Local Search Algorithm for Graph Coloring. *Technical Rapport*, LG2IP/EMA-EERIE, Parc Scientifique Georges Besse F-30000 Nime, France.
- [13] F. Comellas and J. Ozón, An Ant Algorithm for the Graph Colouring Problem, *Proc. ANTS'98- From Ant Colonies to Artificial Ants: First International Workshop on Ant Colony Optimizstion*, Brussels, Belgium, October 1998.
- [14] M. Dorigo and G. Di Caro, The Ant Colony Optimization Meta-heuristic, *Technical Report* , IRIDIA, University of Brussels, 1999.
- [15] A. Caminada, Résolution du Problème de l'Affectation des Fréquences Par Programmation Par Contraintes, *Technical Report FT*, CNET/BEL/POH/CDI/7195/CA, CNET, Belfort, 1995.
- [16] M. Duque Antòn, D. Kunz and B. Rber, Channel Assignment for Cellular Radio Using Simulated Annealing, *IEEE Transactions on Vehicular Technology*, 1993, Vol. 42, 14-21.
- [17] N. Funabiki and Y. Takefuji, Neural Network Parallel Algorithm for Channel Assignment Problems in Cellular Radio Network, *IEEE Transactions on Vehicular Technology*, 1992, Vol. 41, 430-437.
- [18] D. Kunz, Channel Assignment for Cellular Radio Using Neural Networks, *IEEE Transactions on Vehicular Technology*, 1991, Vol. 40, 188-193.
- [19] L. Idoumghar, R. Schott and J.Y Greff, Using Markov Decision Processes to Solve the Frequency Assignment Problem. *Proc. of ASC'2002, 6th IASTED International Conference on Artificial Intelligence and Soft Computing*, 2002, 552-557, ACTA Press.
- [20] Recommendation ITU-R P. 370-7, VHF and UHF Propagation Curves for Frequency Range from 30 MHz to 1000 MHz. *Broadcasting Services*, 1995.
- [21] J.M. Coupé, Using a Tabu Search to Solve the Frequency Assignment Problem, *Patent TDF*, N. 99 02768, 1999.
- [22] F. Glover, Heuristics for integer programming using surrogate constraints, *Decision Sci.*, 8, 1997, 156-166.
- [23] F. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publishers, 1997.



Lhassane Idoumghar is a PhD student at the University of Henri Poincaré - Nancy 1, France. He joined TDF-C2R Broadcasting and wireless Research Center in February 2000. His research activities include the frequency assignment problem, planning processes, stochastic processes, meta-heuristics and applications in computer science.



Philippe Debreux joined TDF C2R in 1994 as a PhD student on Automatics and Signal Processing in collaboration with the University Henri Poincaré - Nancy 1, France. In 1997, after his PhD, he joined the Network and Sites Engineering team of TDF C2R. Since, his activities focus on frequency assignment problems, planning processes. He's involved in international groups for the revision of the plan of broadcasting bands (Stockholm agreement 1961) that will favour the migration to digital sound and video broadcasting.

Two Distributed Algorithms for the Frequency Assignment Problem in the Field of Radio Broadcasting

Lhassane IDOUMGHAR and René SCHOTT

Abstract—The frequency assignment problem involves the assignment of discrete frequencies to the transmitters of a radio network, such as a radio broadcasting network. Frequency separation is necessary to avoid interference by other transmitters to the signal received from the wanted transmitter at the reception region. Here, it is of major importance to minimize the interference while at the same time using the spectrum efficiently.

In this paper we present two original distributed algorithms implemented on clusters of PCs used to solve the frequency assignment problem in the field of radio broadcasting. The first one is based on the island distributed[1] implementation of our hybrid genetic algorithm [2]. The second one uses a distributed cooperative Tabu Search.

Experimental results show that our algorithms, applied to several instances given by TDF-C2R, lead to important time performance improvements.

Index Terms—Parallel Genetic Algorithms, Cooperative Tabu Search Algorithm, Hybrid Algorithms. Problem, Frequency Assignment Problem, Radio Broadcasting.

I. INTRODUCTION

In order to perform suitable geographic and/or demographic coverages for radio broadcasting or radio communications, it is necessary to distribute large numbers of transmitters over the geographic areas. To each transmitter is assigned one frequency. Unfortunately, to each diffusion system¹ is assigned a finite frequency spectrum, which makes it necessary to reuse frequencies. This problem generates interference phenomena due to same or adjacent frequencies allocated to neighbouring transmitters. Here, it is of major importance to minimize the interference while at the same time using the spectrum efficiently. This problem, known to be NP-Hard, can be stated as finding an optimal assignment of frequencies to a set of transmitters, under various constraints.

Due to the NP-Hardness of this problem [3] heuristic methods must be used for large graphs. Many heuristic and metaheuristic methods have been developed in the literature:

- [4], [5], [6], [7], [8] present some constructive algorithms and some basic/notable local searches.
- [9], [10], [11], [12] present some approaches based on genetic algorithm paradigm.
- [13], [14], [8] present an Approximate Non deterministic Tree Search (ANTS) algorithm.

L. IDOUMGHAR is with the LMIA - Université de Haute Alsace, 4 rue des Frères Lumière, 68093 Mulhouse, France. Email: lhassane.idoumghar@uha.fr.
R. Schott is with the IECN & LORIA, Campus scientifique, BP 239, F-54506 Vandoeuvre-lès-Nancy, France. E-mail: Rene.Schott@loria.fr.

¹radio broadcasting, cellular phone networks, etc.

- [15], [16] present a guided local search (GLS) algorithm.
- [17], [18], [19], [20], [21] present some approaches based on Tabu Search algorithm.
- [2], [22], [23], [24] present some hybrid algorithm.
- [25], [26] presents a approach based on ant colony paradigms.

In this paper, we present two original distributed algorithms for solving the frequency assignment problem. The first one implements our Hybrid Genetic algorithm [2] on clusters of PCs. The second one uses a distributed cooperative Tabu Search. Applications of these new algorithms to real instances of frequency assignment problems provided by TDF-C2R Broadcasting and Wireless Research Center show drastic improvements of run time and performances.

This paper is organized as follows: Section 2 describes the radio network planning process. Section 3 recalls the frequency assignment problem. Section 4 gives some hints on Genetic and Tabu Search algorithms. In Section 5 we present our distributed algorithms. Experimental results are given in Section 6. Concluding remarks are contained in the last section.

II. RADIO NETWORK PLANNING

As the demand for telecommunication services increases (interactive TV, mobile phones, FM radio, etc.), providers need fast and efficient design tools to help guarantee their commercial success. By using such tools the intent is to reduce the planning costs and shorten the duration of network deployment by employing various optimisations. Keeping the required hardware investments to a minimal level while achieving a high quality of service is the basic principle of network planning.

The planning process can be split into two main steps. The first one, called *dimensioning*, consists in spreading an optimal number of transmitters over an area in order for this area to be efficiently covered. This number should be as small as possible for obvious economic reasons. Transmitters are spread over the geographical area where providers wish to provide the users with their services, each transmitter covers a part of this geographical area called its *coverage area* (Figure 1). The area around a transmitter where transmission conditions are favourable enough to have a good receipt of the signal is known as the *service area* of the transmitter. The service area is the portion of the coverage area that is not jammed by other transmitters. The transmission quality and thus the

shapes of these regions will depend heavily on the propagation conditions and current interferences produced by the other transmitters. Therefore, coverage areas are frequently of highly irregular shape.

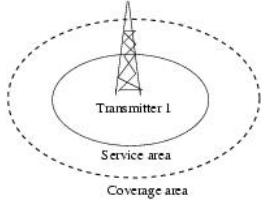


Fig. 1. Coverage and service area of a transmitter

Let W be a set of transmitters distributed across the geographical area. We build a constraints graph as follows:

- 1) apply a propagation routine to each element of W ,
- 2) for each pair of elements in W , compute the intersection of the two corresponding coverage areas,
- 3) each element in W defines a vertex of the graph.

There exists an edge between two vertices if the overlapping of the two corresponding coverage areas is too significant to be ignored. In Figure 2 at interference area the signal sent by transmitter 2 interferes with the one sent by transmitter 1. An example of a constraints graph of the radio network given in Figure 2 is given in Figure 3.

The weight of each edge depends on the kind of radio network. For example, in radio broadcasting the weight assigned to an edge stands for the gap between the frequencies assigned to the vertices at the two endpoints. For example, it is represented by an integer value ranging from 1 to 5. Consequently, at the end of the dimensioning step we are provided with a weighted graph.

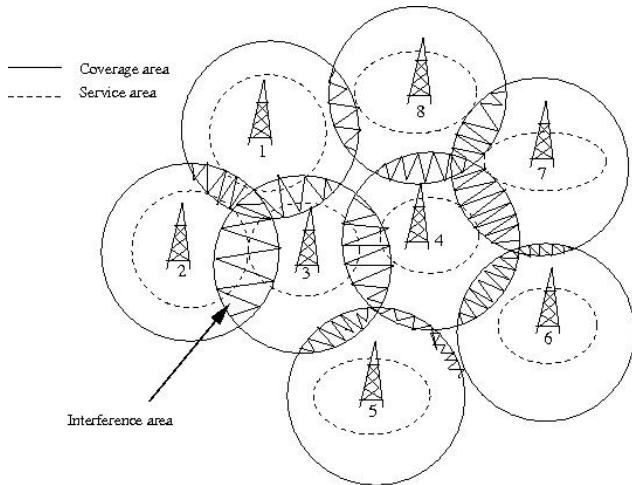


Fig. 2. Influence of a transmitter beyond its service area

In the second step, which is the *frequency assignment* itself, we look for a best frequency to be assigned to each transmitter. This problem will be described in the next section.

III. THE FREQUENCY ASSIGNMENT PROBLEM

The frequency assignment problem can be represented as a generalised graph colouring problem [27]. To formally define

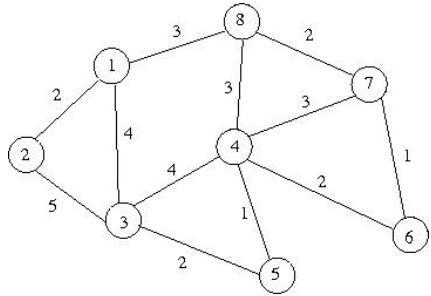


Fig. 3. Constraints graph.

the frequency assignment problem we need some definitions:

- $G=(W, E)$: an undirected graph, where $W = \{w_1, \dots, w_n\}$ is the set of vertices. Each vertex corresponds to a transmitter of the original frequency assignment problem. $E = \{(w_i, w_j) / w_i, w_j \in W\}$ is the set of edges. Each edge of this graph represents a constraint on the frequencies assigned to the two transmitters connected by this edge. An example of constraints graph is given in Figure 3.
- C : set of constraints. $c_{i,j}$ with $i \neq j$ represents the minimum frequency separation required to satisfy the constraint between vertices w_i and w_j . If we indicate with f_i the frequency assigned to transmitter w_i , then if $|f_i - f_j| > c_{i,j}$, the interference involving the two transmitters is acceptable.
- F : set of consecutive frequencies available for every transmitter in G .
- I : is a cost to be paid if the separation between the frequencies of transmitters w_i and w_j is less than or equal to $c_{i,j}$.

Formally, the frequency assignment problem is a mapping $f : W \rightarrow F$ (where F is a set of consecutive integers $0, \dots, k$ representing frequencies) such that the constraints $|f_i - f_j| > c_{i,j}$ are satisfied for all $(w_i, w_j) \in E$. A feasible solution of this problem is represented by a vector $\vec{F}(f_0, f_1, \dots, f_k)$ where $f_i \in F$ for all $i \in [0, k]$. The components of \vec{F} are essentially the frequencies assigned to the vertices of G . This solution must satisfy all constraints if possible, otherwise the number of violated constraints must be minimized. In this case, our priority is to satisfy all highest level constraints.

IV. BACKGROUND

A. Genetic algorithms

Genetic algorithms were developed by Holland [28] to study the adaptative process of natural systems and to develop artificial systems that mimic the adaptative mechanism of natural systems. A review of their theoretical and practical aspects can be found in [29].

Recently, genetic algorithms have been successfully applied to various optimization problems, such as the frequency assignment problem [2]. Genetic algorithms differ from traditional optimization methods in the following ways.

- 1) Genetic algorithms use a coding of the parameter set rather than the parameters themselves.

- 2) Genetic algorithms search from a population of search solutions instead from a single one.
- 3) Genetic algorithms use probabilistic transition rules.

A genetic algorithm consists of a string representation (“genes”) of the solutions (called individuals) in the search space, a set of genetic operators for generating new search individuals, and a stochastic assignment to control the genetic operators.

Typically, a genetic algorithm consists of the following steps.

- 1) Initialization - an initial population of the search solutions is randomly generated.
- 2) Evaluation of the fitness function - the fitness value of each individual is calculated according to the fitness function (objective function).
- 3) Genetic operators - new individuals are generated randomly by examining the fitness value of the search individuals and applying the genetic operators to the individuals.
- 4) Repeat steps 2 and 3 until the algorithm converges.

From the above description, we can see that genetic algorithms use the notion of survival of the fittest by passing “good” genes to the next generation of strings and combining different strings to explore new search solutions. The construction of genetic algorithm for any problem can be separated into four distinct and yet related tasks.

- 1) The choice of the representation of the strings,
- 2) The determination of the fitness function,
- 3) The design of the genetic operators, and
- 4) The determination of the probabilities controlling the genetic operators.

Each of the above four components greatly affects the solution obtained as well as the performance of the genetic algorithm.

In Section 5, we examine each of them for the frequency assignment problem.

B. Tabu Search Algorithm

The basic idea of the *tabu* search metaheuristic [18] is to explore the *search space* of all feasible solutions by a sequence of moves. A move from one solution to another is generally the best available. However, in order to prevent oscillation and to provide a mechanism for escaping from locally optimal but not globally optimal solutions, some moves, at one particular iteration, are classified as forbidden or *tabu*. Moves are regarded as tabu by consideration of short-term and long-term history of the sequence of moves. A very simple use of this idea might be to classify a move as tabu if the reverse move has been made recently or frequently. There is also an *aspiration criterion*, which overrides the tabu moves in particular circumstances. These circumstances might include the cases when, by forgetting that a move is tabu, a solution which is the best so far is obtained.

Suppose it is required to minimize some cost function F on the search space S . For combinatorially hard problems it may only be possible to obtain sub-optimal solutions, in which F is close to its minimum value. Sub-optimal problems may be

Distributed HGA algorithm running on each processor

- Initialize $p_m, p_c \in [0,1]$, $K > 0$, $\text{maxGen} > 0$, $\text{cpt} \leftarrow 0$ and $i \leftarrow 1$
- Construct constraints graph
- Generate subpopulation P_0
- Evaluate P_0 and find the best solution π^*
- $\pi_{\text{Elite}} \leftarrow \pi^*$
- **Repeat**
 - $P_i \leftarrow \emptyset$
 - **For** $j := 1$ to $\text{PopSize}/2$ **do**
 - * Select two parents p_1 and p_2 from P_{i-1} offspring $\leftarrow (p_1, p_2)$
 - * With probability p_c , perform offspring := crossover(p_1, p_2)
 - * With probability p_m , mutate offspring by using the *Tabu Search Algorithm*
 - * Evaluate offspring and add it to P_i
 - Add P_{i-1} to P_i
 - **if** Migrate_condition **then** Receive n Individuals and add its to P_i
 - Sort P_i
 - Keep the PopSize best solution in P_i
 - Find the best solution π^* in P_i
 - Stop criteria
 - * if $\pi_{\text{Elite}} = \pi^*$ then $\text{cpt} \leftarrow \text{cpt} + 1$
 - * π^* is better than π_{Elite} then $\text{cpt} := 0$ and $\pi_{\text{Elite}} \leftarrow \pi^*$
 - * if $\text{fitness}(\pi_{\text{Elite}}) = 0$ OR $\text{cpt} = K$ OR $i = \text{maxGen}$ then terminate HGA
 - $i \leftarrow i + 1$

Fig. 4. Genetic Algorithm that uses TS approach as mutation operator

obtained when a certain threshold for an acceptable solution has been achieved or when a certain number of iterations have been completed.

A characterisation of the search space S for which Tabu Search can be applied is when there exists a set of k moves $D = \{d_1, \dots, d_k\}$ such that the application of the moves to a feasible solution $s \in S$ leads to k , usually distinct, solutions $D(s) = \{d_1(s), \dots, d_k(s)\}$. The subset $N_{\text{set}}(s) \subseteq D(s)$ of feasible solutions is known as the neighbourhood of s .

The method starts with (possibly random) solution $s_0 \in S$ and determines a sequence of solutions $s_0, s_1, \dots, s_n \in S$. At each iteration, s_{j+1} ($0 \leq j < n$) is selected from the neighbourhood $N_{\text{set}}(s_j)$. The selection process is first to determine the tabu set $T_{\text{set}}(s_j) \subseteq N_{\text{set}}(s_j)$ and the aspiration set $A_{\text{set}}(s_j) \subseteq N_{\text{set}}(s_j)$. Then s_{j+1} is the neighbour of s_j which is either an aspirant or not tabu and for which $F(s_{j+1})$ is minimal; that is, $F(s_{j+1}) \leq F(s_i)$ for all $s_i \in (N_{\text{set}}(s_j) - T_{\text{set}}(s_j)) \cup A_{\text{set}}(s_j)$.

In the next section we will present different elements used by our algorithm based on a refined modeling of the input data and on a *tabu search*.

V. OUR DISTRIBUTED ALGORITHMS

A. Distributed Hybrid Genetic Algorithm

The principles of our hybrid algorithm are described in Figure 4.

We use independent subpopulations of individuals with their own fitness functions which evolve in isolation, except for an exchange of some individuals (migration). A set of m individuals is assigned to each of the P processors, for a total

population size of $m \times P$. The set assigned to each processor is its subpopulation.

The processors are connected by an interconnection network with a ring topology. Initial subpopulations consist of a randomly constructed assignment created at each processor. Each processor, disjointly and in parallel, executes the sequential hybrid genetic algorithm on its subpopulation for a certain number of generations ($maxGen$). Afterwards, each subpopulation exchanges a specific number of individuals (migrants) with its neighbours. We exchange the individuals themselves, *i.e.* the migrants are removed from one subpopulation and added to another. Hence the size of the population remains the same after migration. The process continues with the separate evolution of each subpopulation for a certain number of generations. At the end of the process the best individual that exists constitutes the final assignment.

1) Description of an individual: Each individual represents a possible solution to the problem and is composed of a string of genes. In our implementation a gene is coded by an integer array. The i^{th} value corresponds to the frequency assigned to the vertex w_i in a constraint graph. In such graph, some vertices (transmitters) would have fixed frequencies. These vertices are placed at the end of the string. The size of each individual is $|W|$.

2) Initial subpopulation: Initial subpopulation P_0 at each processor is created randomly. In Figure 4, $PopSize$ is the size of every subpopulation P_i . During each of the $maxGen$ generations, $PopSize$ offsprings are generated through the crossover of parents selected from the subpopulation.

3) Fitness evaluation: The fitness function in hybrid genetic algorithms is typically the objective function that we want to minimize in the problem. It serves for each individual to be tested for suitability to the environment under consideration. Our objective function is defined as follows: $I = \sum_{i,j} \rho_{i,j} \delta_{i,j}$, where $\rho_{i,j}$ is a weight associated with constraint $m_{i,j}$ ($\rho_{i,j}$ and $m_{i,j}$ are positive integers). $\delta_{i,j}$ is defined by:

$$\delta_{i,j} = \begin{cases} 0 & \text{if } |f(w_i) - f(w_j)| \geq m_{i,j} \\ m_{i,j} - |f(w_i) - f(w_j)| & \text{otherwise} \end{cases}$$

4) Selection operator: In our approach, we use a Roulette-wheel selection that is a simple method for implementing fitness-proportionate selection. It is conceptually equal to giving each individual a slice of a circular roulette wheel equal in area to the individuals fitness [30]. The wheel is spun N times, where N is the number of the individuals in the population (in our case $N = PopSize$). On each spin, the individual under wheels marker is selected to be in the pool of parents for the next generation [30].

5) Crossover operator: The crossover is a random process defined by a probability p_c and applied sequentially to pairs of parents chosen randomly in the population. It consists in exchanging parts of the genetic material of the parents in order to create two childs (offspring).

In our approach we have used the uniform crossover [31] obtained from a binary mask which possesses a number of genes equal to the number of genes of the individuals of

Sequential Tabu Search Algorithm

- Let X be an initial solution and let $X^* \leftarrow X$ be a best solution
- stop \leftarrow false, tabu_list $\leftarrow \emptyset$,
- **While** not stop **Do**
 - Let s^* be the best neighbouring configuration which is not forbidden by the tabu_list. This configuration is chosen by using a *uniform probability* in *Skiplist*.
 - Choose s_t in the best neighbouring configuration forbidden by tabu_list (Aspiration criterion).
 - Choose the best configuration X between s_t and s^* which minimizes a cost function.
 - Adjust tabu_list
 - Update the interferences matrix M .
 - **If** X is better than X^* **Then** $X^* \leftarrow X$
 - stop \leftarrow termination criterion

Fig. 5. Tabu Search approach used as a mutation operator

the population. This mask is usually uniformly randomly generated: each bit has value 0 or 1 with equal probability. The used uniform crossover is parameterized by a function of the probability $p_0 = 0.5$ [32], corresponding to appearance of the values (0 and 1) in each bit of the mask. The first child is created by taking the genes of the first parent corresponding to value 1 in the mask. The second child is obtained similarly but the complementary part of the mask is used.

6) Mutation operator: As indicated previously, we use *Tabu Search Algorithm* described in Figure 5. This algorithm implements some particular features, which are not present in the general schema and which we list below. These features improve the performances of our Tabu Search Algorithm. Namely:

- **Neighbouring configuration:** an solution X' is in the neighbourhood of the current solution X if X' differs from X in the frequency assigned to exactly one transmitter. *Skiplist*² data structure [33] allows us to choose efficiently and quickly the transmitter who violates more constraints. To do this, all transmitters are ranked in descending order of their *local cost function*³. *Skiplist* contains pairs $< I_{local}(w_i), w_i >$. The transmitter for which we must change the frequency is the first transmitter w in *skiplist*. For this transmitter we choose the best assignment f_{best} by using the *interferences matrix* M (see next point). The move $< w, f_{old} >$ which produces X' from X is added to the tabu list and *Skiplist* is updated (remove $< I_{local}(< w_i, f_{old} >), w_i >$ and add $< I_{local}(< w_i, f_{new} >), w_i >$)

- **Local cost function:** to improve performances of our algorithm, especially when we compute the objective function, we have implemented a two dimensional array $|W||F|$, called the *interferences matrix* M inspired by [34], [23] where each element $M(w_i, f_j)$ corresponds to the *local cost (Interferences)* $I_{local}(w_i)$ obtained by assigning the frequency f_j to transmitter w_i . The

²A skip list of size n is a probabilistic data structure, based on multiple parallel, sorted linked lists, with efficiency comparable to a binary search tree (insertion, deletion, query have an average time complexity in $O(\log(n))$).

³ $I_{local}(w_i) = \sum_{w_j \in adjacent(w_i)} \rho_{w_i, w_j} \delta_{w_i, w_j}$

Cooperative Distributed Tabu Search Algorithm

- Let X be an initial solution and let $X^* \leftarrow X$ be a best solution
- stop \leftarrow false, tabu_list $\leftarrow \emptyset$,
- **While** not stop **Do**
 - cooperation_send_receive(best_solution)
 - **If** best_solution is better than X **then**
 - * tabu_list $\leftarrow \emptyset$, $X \leftarrow$ best_solution
 - **else**
 - * Let s^* be the best neighbouring configuration which is not forbidden by the tabu_list. This configuration is chosen by using a *uniform probability* in *Skiplist*.
 - * Choose s_t in the best neighbouring configuration forbidden by tabu_list (Aspiration criterion).
 - * Choose the best configuration X between s_t and s^* which minimizes a cost function.
 - * Adjust tabu_list
 - * Update the interferences matrix M .
 - **If** X is better than X^* **Then** $X^* \leftarrow X$
 - stop \leftarrow termination criterion

Fig. 6. Distributed Tabu Search Algorithm.

best frequency f_{new} to be assigned to a transmitter w_i , corresponds to the frequency which has the smallest $M(w_i, f_{new})$ value. Each time a move $< w_i, f_{new} >$ is chosen, only rows corresponding to all w_j adjacents to w_i , are updated accordingly. More precisely, for each w_j only the positions of $M(w_j, f)$ corresponding to the frequencies f which interfere with the f_{old} or the f_{new} frequency of the transmitter w_i are modified. Thus, the objective function of a current solution X is computed as follows:

$$I_{new}(X) = I_{old}(X) - I_{local}(< w_i, f_{old} >) + I_{local}(< w_i, f_{new} >)$$

- **Aspiration criterion:** the tabu status of a move is cancelled if the move leads to a solution better than the best solution X^* found so far.
- **Tabu list:** the tabu list in our approach contains pairs $< w_i, f >$, where w is a transmitter and f a frequency. Each time a move $< w_i, f >$ is chosen, we insert it into the tabu list and it is classified tabu for l iterations, where l is a tabu list size.
- **Termination criterion:** The algorithm stops when the maximum number of iterations has been exceeded or if there was no improvement of the best solution in the last k iterations.

B. Cooperative Distributed Tabu Search Algorithm

The principles of this algorithm, described in Figure 6, are as follows:

The processors are connected by an interconnection network with a ring topology. Each processor, disjointly and in parallel, executes the sequential Tabu Search Algorithm on its current solution. After each k iterations, it exchanges its best solution (obtained so far) with its neighbours by calling *cooperation* method.

The *cooperation_send_receive* method implements:

- *Synchronous mode (PTSSy):* at the k^{th} iteration, each processor sends its best solution and waits to receive one solution from its neighbour.
- *Asynchronous mode (PTSAs):* at the k^{th} iteration, each processor sends its best solution and continues the improvement of its current solution and verifies if it does not receive a solution from its neighbour.

The process continues with the separate improvement of each current solution for a maximum number of iterations. At the end of the process the best solution that exists constitutes the final assignment.

VI. EXPERIMENTAL RESULTS

The island distributed hybrid genetic and Tabu Search algorithms have been implemented on a cluster of PCs running under Windows XP Professional version 2002. The cluster is composed of 5 Pentium (D) machines running at 3 Ghz. Each processor has 1 Gb of memory.

The results presented in this paper have been achieved with the machines running their normal daily loads in addition to our algorithm.

Each approach described in this paper is written in C++ using MPICH2 (version 1.0.6) library for communication across the processes. The C++ compiler used is gcc version 2.95.2 (Dev-cpp).

This section presents the main results obtained after several runs performed over three real instances, which can be obtained from the corresponding author, of frequency assignment problems provided by TDF-C2R Broadcasting and Wireless Research Center:

- *instance1:* 972 transmitters and 12933 constraints,
- *instance2:* 972 transmitters and 25550 constraints,
- *instance3:* 972 transmitters and 43807 constraints.

Each vertex of the constraints graph may take $m = 200$ possible values (bandwidth 87.5-107.5 MHz, $|107.5 - 87.5\text{ MHz}| = 20\text{ MHz} = m \times 100\text{ kHz}$).

We recall that the main objective is to satisfy all constraints (interference constraints) if possible, otherwise high level constraints have to be satisfied first.

The results produced by the approach presented in this paper and those obtained by other approaches [2], [25] compared to the best operating solution in the field of radio broadcasting given by CSA [35] in France are given in Table 1, Table 2 and Table 3.

In these tables:

- *SeqTS:* means that we use *Sequential Tabu Search Algorithm* described in Figure 5.
- *SeqHGA:* means that we use a sequential hybrid genetic algorithm described in Figure 4. This algorithm uses a Tabu Search algorithm described in Figure 5 as a mutation operator.
- *ANT:* means that we use an ant algorithm [25]. This algorithm is a multiagent system based on the idea of parallel search. In this algorithm, a given number of ants move around the vertices of the graph and change the colour of each visited vertex according to a local criterion.

TABLE I
COMPARISON OF THE RESULTS OBTAINED BY DIFFERENT METHODS USED TO SOLVE *instance1*.

	SqHGA	ParHGA	ANT	PTSAs	PTSSy	SeqTS	CSA
Total violated constraints	64	64	70	66	66	68	142
constraints 1	20	23	17	23	21	27	37
constraints 2	39	36	41	37	34	35	68
constraints 3	4	4	9	4	4	4	15
constraints 4	1	1	3	2	2	2	6
constraints 5	0	0	0	0	0	0	16
run time	17 sec	6 sec	-	2 sec	2 sec	2 sec	-

TABLE II
COMPARISON OF THE RESULTS OBTAINED BY DIFFERENT METHODS USED TO SOLVE *instance2*.

	SqHGA	ParHGA	ANT	PTSAs	PTSSy	SeqTS	CSA
Total violated constraints	81	77	99	82	81	84	271
constraints 1	36	38	47	42	40	43	92
constraints 2	42	35	42	34	38	35	117
constraints 3	2	3	8	3	1	3	27
constraints 4	1	1	2	3	2	3	10
constraints 5	0	0	0	0	0	0	25
run time	27 sec	15 sec	-	3 sec	3 sec	3 sec	-

TABLE III
COMPARISON OF THE RESULTS OBTAINED BY DIFFERENT METHODS USED TO SOLVE *instance3*.

	SqHGA	ParHGA	ANT	PTSAs	PTSSy	SeqTS	CSA
Total violated constraints	104	104	151	114	105	119	466
constraints 1	59	59	74	67	57	68	157
constraints 2	42	41	68	43	44	47	202
constraints 3	2	2	6	2	2	2	44
constraints 4	1	2	3	2	2	2	25
constraints 5	0	0	0	0	0	0	38
run time	1 min 47 sec	17 sec	-	5 sec	5 sec	5 sec	-

TABLE IV
HYBRID GENETIC ALGORITHM : DEFAULT VALUES OF THE PARAMETERS

Population size (PopSize)	200 for the sequential version. 40 for each sub-population in distributed version
Mutation probability m_m	0.5
Crossover probability p_c	0.8
Migration rate	5
Stopping criterion	The stopping criterion used is the non evolution of the best solution during $K = 15$ generations, once the minimal number of generations has been overpassed ($maxGen = 200$).

At each iteration each ant k moves from vertex i to vertex j and changes its colour, it remains there until the next iteration, when it will move again towards one of the vertices of j 's neighbourhood. Each ant can "remember" at each step the former changes performed by the algorithm, and takes into account that these changes may have modified the cost function of the neighbourhood of j . Therefore, at the next step, the ant k will normally try to arrange the colouring of the worst adjacent vertex to j . Any single action depends strongly on the last move of each ant; this dependence reinforces the results of recent modifications. At a given iteration each ant moves from the current vertex to the adjacent vertex with the highest level of violations, and replaces the old colour of the vertex with a new colour that minimizes these violations. For a given vertex, the highest level of violations is computed by using the *Cost* function described in Section 4. This action is randomly repeated for each ant: the ant moves to the worst adjacent vertex with a certain probability p_n (otherwise it moves to any other adjacent vertex randomly chosen), and assigns the best possible colour with a probability p_c (otherwise any colour is assigned at random). The probabilistic nature of the algorithm allows the ants to escape from local minima and to obtain bounds close to the absolute minimum. This process, which is carried out simultaneously by the set of ants, is repeated until the optimal solution is found or the algorithm converges. The number of ants that move along the graph is an adjustable parameter and increases with the order of the graph.

- *PTSSy* and *PTSAs*: means that we use *Synchronous and Asynchronous* Distributed Tabu Search Algorithm described before.
- *ParHGA*: means that we use *Distributed hybrid genetic algorithm* described in Figure 4.
- *CSA*: corresponds to the best operating solution in the field of radio broadcasting given by the CSA [35] in France,
- The second row gives the total number of constraints that are violated by each approach,
- The next five rows give the details of the number of total violated constraints for each constraint type,
- The last row gives the run time of each algorithm.

The analysis of the results given in these tables shows important time performance improvements.

Table 4 gives the default values of the parameters used by the hybrid genetic algorithm.

VII. CONCLUDING REMARKS

We have presented two original distributed algorithms for solving the frequency assignment problem. Experimental results show important time performance improvements. The design of even better hybrid sequential and parallel algorithms for solving this problem is a real challenge.

Acknowledgments.

The authors are very grateful to anonymous referees for their

comments and suggestions which allowed to improve the contain and the presentation of this paper.

REFERENCES

- [1] M. Khabzaoui, C. Dhaenens and E. Talbi, Parallel Genetic Algorithms for multi-objective rule mining, 6th Metaheuristics International Conference (MIC'2005), Vienne, Autriche, pp. 571-576, 2005.
- [2] L. Idoumghar and M. Alabau and R. Schott, New Hybrid Genetic Algorithms for the Frequency Assignment Problem, *Journal of IEEE Transactions on Broadcasting*, March 2002, Vol.48, N.1, 27-34.
- [3] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman and Co., 1979.
- [4] R. Borndorfer, A. Eisingerblatter, M. Grotschel, and A. Martin. Stable set and other techniques for frequency assignment problems. Preprint, Konrad-Zuse-Zentrum fur Informationstechnik Berlin.
- [5] R. Borndorfer, A. Eisingerblatter, M. Grotschel, and A. Martin. Frequency assignment in cellular phone networks. *Annals of Operations Research*, 76, pp. 73-93, 1998.
- [6] A. Eisenblatter. A frequency assignment problem in cellular phone networks (extended abstract). Technical Report SC 97-27, Konrad-Zuse-Zentrum fur Informationstechnik Berlin, July 1997.
- [7] A.M.C.A. Koster. Frequency Assignment. Models and Algorithms. PhD thesis, Maastricht University, November 1999.
- [8] R. Montemanni. Algoritmi Esatti ed Euristicci per il Problema dell'Assegnamento di Frequenze. Tesi di Laurea, Universita degli Studi di Bologna, March 1999. In Italian.
- [9] A.W.J. Kolen. A genetic algorithm for frequency assignment. Technical report, Maastricht University, 1999.
- [10] T.L. Lau and E.P.K. Tsang. Solving the radio link frequency assignment problem with the guided genetic algorithm. In Proceedings of the NATO RTA SET/ISET Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace), Aalborg, Denmark, October 1998, NATO RTO-MP13, pp 14b.1-14b.27, 1999.
- [11] G. Colombo, A genetic algorithm for frequency assignment with problem decomposition. *Int. J. Mobile Network Design and Innovation*, V. 1, No. 2, pp. 102-112.
- [12] S. Matsui, I. Watanabe, and K. Tokoro, An Efficient Genetic Algorithm for a Fixed Frequency Assignment Problem with Limited Bandwidth Constraint, *Systems and Computers in Japan*, Vol. 35, No. 10, pp. 32-39, 2004.
- [13] V. Maniezzo and A. Carbonaro. An ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems*, 16, pp. 927-935, 2000.
- [14] V. Maniezzo, A. Carbonaro, and R. Montemanni. An approach to frequency assignment problem based on an ANTS heuristic. In Proceedings of the Third Metaheuristic International Conference (MIC'99), pp. 311-316, 1999.
- [15] C. Voudouris. Guided Local Search for Combinatorial Optimisation Problems. PhD thesis, University of Essex, 1997.
- [16] C. Voudouris and E. Tsang. Solving the radio link frequency assignment problem using guided local search. In Proceedings of the NATO RTA SET/ISET Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace), Aalborg, Denmark, October 1998, NATO RTO-MP13, pp. 14a.1-14a.9, 1999.
- [17] J.M. Coupé, "Using a Tabu Search to Solve the Frequency Assignment Problem", *Patent TDF*, N. 9902768, 1999.
- [18] F. Glover and E. Taillard, A User's Guide to Tabu Search, *Annals of Operations Research*, 41, 1993, 3-28.
- [19] L. Idoumghar and Ph. Debreux, New Modeling Approach for the Frequency Assignment Problem in Broadcasting, *Journal of IEEE Transactions on Broadcasting*, pp. 293-298, Vol. 48, N 4, December 2002.
- [20] R. Montemanni, D.H. Smith, and S.M. Allen. An improved lower bound for the fixed spectrum frequency assignment problem. *J. of IEEE Transactions on Vehicular Technology*, V.52, No. 4, pp.891 - 901, July 2003.
- [21] R.M. Whitaker, S. Hurley, and D.H. Smith. Frequency assignment heuristics for area coverage problems. Technical report, Cardi University, September 2000.
- [22] L. Idoumghar and R. Schott, A New Hybrid GA-MDP Algorithm for the Frequency Assignment Problem , Proceedings of ICTAI'06 (18th IEEE International Conference on Tools with Artificial Intelligence), Washington, DC, November 13-15, 2006, 18-25, IEEE Pub.
- [23] C. Fleurant and J.A. Ferland, Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research*, 63, pp. 437-461, 1996.

- [24] A. Capone and M. Trubian, Channel assignment problem in cellular systems: a new model and a tabu search algorithm, IEEE Transactions on Vehicular Technology. Vo. 48, No. 4 pp. 1252-1260, July 1999.
- [25] F. Comellas and J. Ozón, An Ant Algorithm for the Graph Colouring Problem, *Proc. ANTS'98- From Ant Colonies to Artificial Ants: First International Workshop on Ant Colony Optimizotion*, Brussels, Belgium, October 1998.
- [26] J. S. Graham , R. Montemanni , J. N. J. Moon and D. H. Smith, Frequency assignment, multiple interference and binary constraints Wireless Networks. Vo. 14, No. 4, pp. 449-464, August 2008.
- [27] S. Hurley and D.H. Smith and S.U. Thiel, Fasoft: A System for Discrete Frequency Assignment, *Radio Science*, 32(5), 1997, 1921-1939.
- [28] J. H. Holland, Adaptation in Natural and Artificial Systems, *PhD thesis*, University of Michigan press., 1974.
- [29] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, *Addison Wesley*, 1989.
- [30] Mitchell, M. (1998). An Introduction to Genetic Algorithms. Massachusetts: The MIT Press.
- [31] G. Syswerda, Uniform Crossover in Genetic Algorithms, *Proc. of Int. Conf. on Genetic Algorithms (ICGA'89)*, pp. 2-9, 1989.
- [32] W.M. Spears and K.A. De Jong, On the Virtues of Parametrized Uniform Crossover, *Proc. of the 4th Int. Conf. on Genetic Algorithms and Their Applications*, University of California, San Diego, Morgan Kauffmann Publishers, pp. 230-236, 1991.
- [33] W. Pugh, Skip List: A Probabilistic Alternative to Balanced Trees. *CACM*, Vol. 33, No.6, pp. 668-676.
- [34] J.-K. Hao, R. Dorne, and P. Galinier, Tabu Search for frequency assignment in mobile radio networks. *Journal of Heuristics*, 4, pp. 47-62, 1998.
- [35] The “Conseil suprieur de l’audiovisuel” is an independent administrative authority which was created in France to guarantee broadcasting freedom in the conditions laid down by the modified Law of September 30th, 1986. “<http://www.csa.fr/multi/introduction/intro.php?l=uk>”



Lhassane IDOUMGHAR is a associate Professsor at the University of Haute Alsace in France since 2003. He obtained his PhD at the University of Henri Poincaré - Nancy 1, France. 2000-2003 he worked in TDF-C2R Broadcasting and wireless Research Center. His research activities include the frequency assignment problem, plannig processes, stochastic processes, meta-heuristics and applications in computer science.



René SCHOTT is a full Professor at the University Henri Poincaré-Nancy 1, France since 1987. He has granted the Silver Medal from the National Center for Scientific Research (CNRS) with the team EURECA, CRIN (Centre de Recherches en Informatique de Nancy) in 1987 for his achievements in theoretical computer science. He is a member of the American Mathematical Society, the French Mathematical Society and the European Association for Theoretical Computer Science. His research interests include the study of stochastic processes and operator calculs on algebraic structures and their applications in computer science (probabilistic analysis of algorithms, random generators, randomized algorithms, optimization problems). More recently, he has been focusing on cellular networks and frequency assignment problems. He serves as referee for many journals and international conferences both in mathematics and in computer science. Dr. Schott is the co-author of seven books and has also published (as author or co-author) over 100 journal and conference proceedings papers. He received numerous collaborative research grants from such agencies as CNRS, INRIA, NATO, NSF and PROCOPE.

Optimized Fuel Cell Array Energy Management Using Multi-Agent Systems

Robin Roche¹, Lhassane Idoumghar², Benjamin Blunier¹ and Abdellatif Miraoui¹

¹ Université de Technologie de Belfort-Montbéliard, Laboratoire Systèmes et Transports, France

² Université de Haute Alsace, LMIA / INRIA Grand Est, France

Corresponding author: robin.roche@utbm.fr

Abstract—Modular fuel cell power plants are a type of energy source which were recently commercially developed. Their main advantages are their limited environmental impact compared to traditional power sources such as coal- and gas-fired power plants, and their ability to be controlled, as opposed to intermittent energy sources. These modules can be used as primary or secondary energy sources and can for example help minimize the power drawn from the grid, then only a backup power source. However, controlling an entire array of fuel cells causes problems similar to the ones encountered by thermal power plant operators: how can the operation of such units be optimized? This article describes a method to help reach a better economic performance for fuel cell arrays in which the individual performance of each fuel cell is considered. The method is based on a hybrid system consisting in a multi-agent system, an expert system and an optimization algorithm for optimally dispatching the load. By enabling the use of storage and the ability to automatically start and stop fuel cells, simulation results show that the method can help minimize running costs as well as enable a very flexible control system.

I. INTRODUCTION

Fuel cells have been a subject of research for several decades, leading to a number of mobile applications as for satellites and hybrid vehicles. However, their commercial applications, especially for stationary uses, are still quite rare today. Reasons for this situation include expensive parts and a low efficiency of the system when hydrogen production and transport are taken into account.

A step forward toward a broader use of fuel cells was done with the development and the commercialization of fuel cells as small power plant modules. The Bloom Energy Server [1] of Bloom Energy is an example. It offers a flexible and modular design which allows to assemble several 100 kW fuel cell modules into an array. The software company Adobe has for instance bought 12 modules to create its own power plant of a rated power of 1200 kW and meet about one third of the electricity needs of its headquarters.

The management and control of such power plants depends a lot on their objectives: if the plant is used as a constant power source, it will probably always run at its best efficiency point, whereas if it is used as a secondary source, the most economically sound choice may be different. Energy management of fuel cell arrays is rarely treated in the literature, except for a few examples as Fairfax and Dowling [2], who describe the

operation of a couple of 200 kW fuel cells as complements to other primary power sources during demand peaks.

The following sections describe the energy management system of a fuel cell array-based power plant. A first section shows how the elements are modeled, then how the multi-agent system is used, how the energy management algorithm works and finally what results and benefits several simulations give.

II. SYSTEM MODELING

The system studied in this work consists in several fuel cells, a storage unit, a distribution power line and an aggregated load (figure 1). As the focus is set on energy management and not on modeling, the following models are purposely kept simple. They are primarily used to obtain the cost function of each component, which will be used in the optimization process. As a consequence, the proposed method has some similarities with model-based control.

A. Fuel cell model

The fuel cell we modeled has a rated stack power $P_{s,r}$ of 100 kW, similarly to Bloom Energy's modules. A Ballard Mark-V cell [3] was used as a basis for sizing the whole module as a proton exchange membrane fuel cell (PEMFC).

The polarization curve $V_c(j)$ of the cell gives its voltage as a function of the current density j . This curve is used to compute the value of j_M at which the power output $P_{c,M}$ of the cell is maximal as well as the corresponding cell voltage $V_{c,m}$. From these values and from the expected minimal stack voltage $V_{s,m}$, we can deduce the number of cells n_c required in the stack:

$$n_c = \frac{V_{s,m}}{V_{c,m}} \quad (1)$$

The total active surface S_t of the cell is then derived:

$$S_t = \frac{P_{s,r}}{P_{c,M}} \quad (2)$$

Current I is obtained from the previous terms in (3). From this, we deduce the expression of $V_s(I)$, the voltage of the fuel cell stack, and $P_s(I)$, its power output:

$$I = \frac{j \cdot S_t}{n_c} \quad (3)$$

$$V_s(I) = n_c \cdot V_c(I) \quad (4)$$

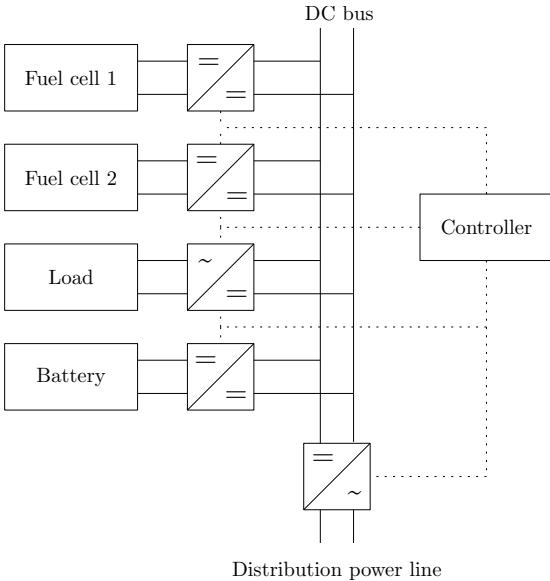


Fig. 1. The studied system consists in several fuel cells, a storage unit, a distribution power line and a load. These elements are controlled by agents located in power converters and a central control agent. Dotted lines correspond to communication channels.

$$P_s(I) = V_s(I) \cdot I \quad (5)$$

To take into account losses due to the auxiliaries of the fuel cell, such as the air compressor, two types of losses were introduced when computing the net power output in (6). The variable $\eta_{l,a}$ represents the share of P_s consumed by a part of the auxiliaries and $P_{l,a}$ the constant power consumed by others.

$$P_{f,n}(I) = (1 - \eta_{l,a})P_s(I) - P_{l,a} \quad (6)$$

The mass flow m_{H_2} of hydrogen, i.e. the fuel consumption of the fuel cell, is obtained using the following relation where M_{H_2} is the molar mass of hydrogen and F is the Faraday constant:

$$m_{H_2} = M_{H_2} \frac{n_c \cdot I}{2F} \quad (7)$$

The hydrogen consumption of the fuel cell $m_{H_2}(P_{f,n})$, shown in figure 2, can finally be used to compute the cost of each fuel cell for any set point within its working range.

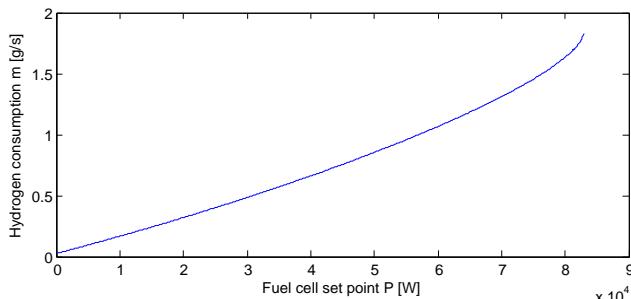


Fig. 2. Hydrogen consumption of the fuel cell model

B. Battery model

The storage model we used represents a battery but could also be used with other types of storage due to its generic nature. Its characteristics are its capacity C_b , its charge and discharge efficiency η_b , its initial state of charge (SoC) and its maximal power output. The minimal and maximal SoC bounds, outside which the unit should not be used without being strongly deteriorated, are also defined. The SoC is updated using (8), where P_b is the set point and Δt the duration between two set points:

$$SoC_t = SoC_{t-1} + \eta_b \frac{P_b \cdot \Delta t}{C_b} \quad (8)$$

C. Power line model

This component is used to buy and sell energy to the grid and is modeled as a simple power line with a maximum capacity. Variable price profiles can be defined and are used to compute the corresponding energy costs.

III. MULTI-AGENT SYSTEM

The energy management algorithm used in this work is based on a multi-agent system. This approach is selected to enable a high degree of flexibility in the control system, as well as an approach to a distributed control system in which the various components of the microgrid are clearly separated and have their own capabilities. The following explains what agents are and why and how they are used.

A. Concept

A multi-agent system (MAS) consists in a group of entities called agents. These agents can communicate with each other and perceive the environment in which they exist and interact with it [4], as shown in figure 3.

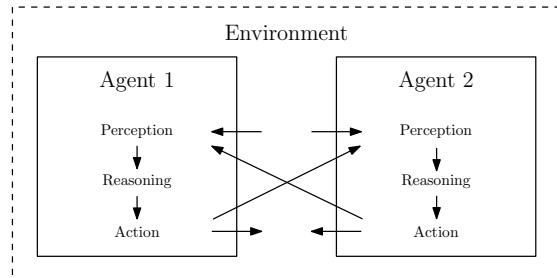


Fig. 3. A multi-agent system consists in several entities called agents which can perceive their environment and interact with it

Agents can also exhibit various degrees of intelligence. A first category of agents corresponds to agents which have simple thought processes and are called reactive agents. In most cases, they just react to an external stimulus from the environment in a way defined by the programmer. Although these agents remain simple, they can in some cases lead to a very powerful collective intelligence when they can interact with other similar agents, in a process called emergence.

On the other hand, another category of agents called cognitive agents can have much more complex reasoning

capabilities. Several models were developed over the years, such as the BDI (Beliefs, Desires and Intentions) model [5]. These agents can for example take decisions based on their internal representation of the environment and establish plans and schedules to reach their goals.

B. MAS for power systems

The evolution of existing power grids toward smarter grids partially results from the increase of the share of distributed energy resources such as wind turbines and solar panels. This trend is a move from centralized to distributed power systems and requires control systems adapted to the new challenges it creates. MAS suit well the needs of these new grid architectures for the following reasons [6]–[8]:

- *They are distributed:* MAS are inherently distributed in their nature and can therefore be easily adapted to control distributed sources, storage and loads. This structure also enables agents to have a local view of their environment, as having information on the whole grid would be useless for them. It also means that the amount of data to be transferred between agents can be limited, as well as the computing power required to treat this data.
- *They are intelligent:* Agents, whether they are reactive or cognitive, are able to communicate with each other to get the information they need. This information can then be used in various processes, to take decisions or to make plans. In this work, the reasoning capability of agents is for example used to decide if some fuel cells need to start or stop, and to find the optimal set points which minimize the total generation cost.
- *They are flexible and fault tolerant:* As the structure of a power grid changes every time a source or a load is connected or disconnected, the control system needs to take these changes into account. In a MAS, a failure in a source or a power line is just seen as a modification of the architecture of the grid and the MAS can try to find alternative solutions, even if the event was not initially predicted by the programmer. Similarly, components connected to the grid “on-the-fly” can also be controlled by the MAS.

C. Problem agentification

The system studied in this work was transformed into a MAS, or “agentified”. Six types of agents are used:

- *Control agent:* This is the central agent in the microgrid. It retrieves information on the components connected to the grid and computes the optimal set points to send to the sources, the storage units and the grid agents. Its internal working is summarized in figure 4. This agent can be considered as intelligent, as opposed to others which are rather reactive.
- *Fuel cell agents:* Each fuel cell, based on the previous model, is controlled by an agent which is able to send data on the state of the fuel cell and to receive set points from the control agent in return.

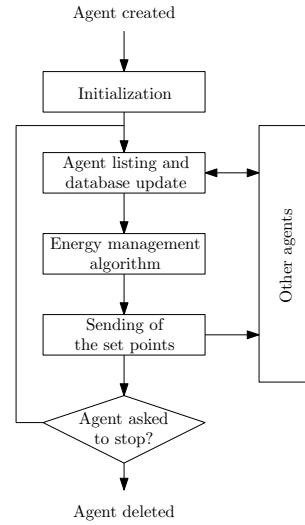


Fig. 4. The control agent is an intelligent agent which plays a central role in the system and interacts with other agents. It holds a database of the agents and their properties and runs the energy management algorithm.

- *Storage agents:* Similarly to the fuel cell agents, these agents control the storage units, here batteries.
- *Grid agents:* These agents inform the control agent of the price at which power is being bought or sold and receives the value of how much it can sell or buy.
- *Load agents:* Each load agent acts as an observer which sends the control agent how much the load consumes and how much it expects to consume in the near future.
- *GUI agent:* This agent serves as a graphical user interface (GUI). It shows the user how the system evolves and it allows him to virtually control, connect or disconnect components, even when a simulation is running.

In reality, these agents would be connected to a common telecommunication network to communicate with each other. The control and GUI agents would be installed on a computer and the others would directly control the current-controlled power electronics converters attached to the components they represent, as in figure 5 [9].

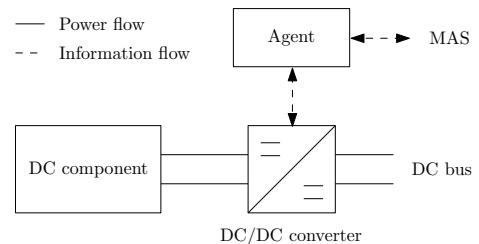


Fig. 5. In reality, agents would control the converters attached to the components, such as fuel cells and batteries

D. MAS implementation

The MAS was implemented using a multi-agent middleware called JADE, for Java Agent DEvelopment framework [10].

JADE provides a framework to develop agents which allows programmers to focus on developing the agents themselves. It respects FIPA [11] standards, especially for communication between agents, and has several tools to help programmers such as a directory of agents, a message examiner, etc.

The life-cycle of each agent starts with its initialization. The agent registers in a database of existing agents so that it can be found by the control agent. Each agent then continuously runs its behavior(s) (the roles of the agents described in the previous section) until a stop condition is met. For example, the behavior of a fuel cell agent includes sending its state data to the control agent and receiving set points from it. It can also adapt to the changes in the fuel cell's state (stopped, starting, running or stopping). Finally, when the agent is about to be deleted, some clean up instructions are run such as de-registering from the list of running agents.

IV. ENERGY MANAGEMENT ALGORITHM

A. Hypotheses

The proposed algorithm (figure 6) aims at dispatching active power, by determining its share at each node of the grid while minimizing the overall generation cost and verifying constraints on active power. The following assumptions are used:

- Other algorithms and devices are in charge of stability, safety and voltage control. The grid, with its buses, converters and safety equipment is not modeled. Reactive power and voltage control are considered as separate problems, and do not significantly impact energy management in terms of costs.
- The fuel cells are the primary sources to power the load. The storage units are only used when the fuel cells cannot meet demand. Similar conditions apply to buying and selling energy from the main grid.
- The fuel cells are heterogeneous. Some are older or were used more intensively than others and have degraded efficiency, which is translated in the algorithm by a coefficient applied to the cost of the fuel cells. An older fuel cell for example consumes more to produce the same output as a recent one. The fuel cells' dynamic is also voluntarily slowed down using ramps to limit their aging.

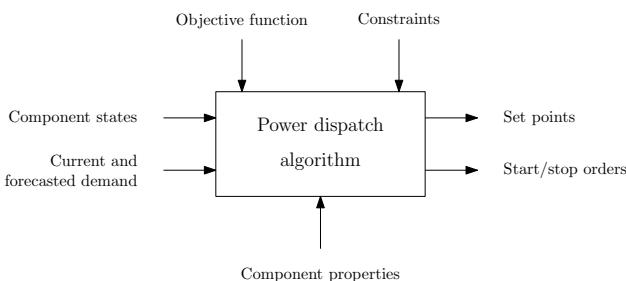


Fig. 6. A summary of the inputs and outputs of the power dispatch algorithm

B. Dynamic economic dispatch

Solving the economic dispatch problem is achieved in two steps. The first one is based on sets of rules from an expert system. These rules determine what will be the working ranges of the components connected to the grid. This is primarily used to restrain the size of the search space of the second step, in which the optimization algorithm tries to find the optimal solutions to the current grid configuration. This methodology corresponds to a form of dynamic optimization, where the next solution depends on the previous one and the constraint changes at every round. A faster convergence is enabled by not going through the whole search space: it makes no sense to search in a space where the components cannot physically go within the given time constraint.

C. Rule-based expert system

The dispatch algorithm is firstly based on a set of rules which define the power bounds in which the components (fuel cells, storage units and grid) can operate. These rules set general tendencies that reflect some of the objectives and constraints of the operation of the power plant. The following describes a simplified version of the actual rules.

A first set of rules is for the fuel cells. They take into account the various modes in which the turbines can operate:

- In starting mode, the fuel cells have no output.
- In running mode, the bounds are set with respect to the dynamics of the fuel cells, simplified as ramp down R_d and ramp up R_u constraints, set to 500 W/s:

$$P_t - P_{t+1} \leq R_d \quad (9)$$

$$P_{t+1} - P_t \leq R_u \quad (10)$$

- In stopping mode, the bounds are set so that the fuel cells stop as quickly as possible while respecting their dynamic constraints.

The use of storage units depends on whether the fuel cells can satisfy demand and if the SoC of the units is too close to their lower and higher limits. If these conditions are verified, then the storage units are not used. Otherwise:

- If demand is too high for the fuel cells and the SoC is high enough, the units are allowed to discharge,
- If demand is too low and the SoC is low enough, the units are allowed to charge.

Finally, selling and buying energy from the grid is controlled similarly to a storage unit, and is mostly used when the storage unit is out of order or if the prices are competitive enough.

A simple example of how these rules are used is shown in figure 7. The first fuel cell has already reached its maximum output power and can only maintain or decrease it. The second one is subject to the two ramp-up and ramp-down constraints described earlier, meaning that the optimization algorithm will only search over a range of 1 kW in this dimension. The third and last fuel cell is just starting and is constrained to a nil output. The storage unit can handle much faster dynamics and the corresponding optimization range is thus much bigger

(100 kW in this example). This may happen when a sudden fall in generation or demand is detected.

Component	P_t	$P_{t+1,\min}$	$P_{t+1,\max}$
Fuel cell 1	83.0	82.5	83.0
Fuel cell 2	53.5	53.0	54.0
Fuel cell 3	0	0	0
Storage	0	0	100.0

Fig. 7. Examples of optimization bounds. The minimum and maximum bounds are computed based on the current set points and the maximum dynamics the components can handle. Values are displayed in kW.

D. Optimization algorithm

The role of the optimization algorithm is to compute the optimal set points to send to the components of the system. After a short comparison with other common optimization techniques, such as genetic and evolutionary algorithms, an algorithm based on particle swarms was selected for its performance.

A particle swarm optimization (PSO) algorithm [12] is a population-based stochastic optimization technique inspired by the social behavior of animals, such as bird flocking. Each individual of the population, or particle, tries to improve itself by observing other group members and imitating the best ones. To do that, each particle keeps track of its coordinates in the solution space using two values: the best solution it has achieved so far, called personal best p_{best} , and the best value obtained so far by any particle in the neighborhood of that particle, called group best g_{best} .

A standard PSO algorithm follows the following process for each particle, after it has been initialized:

- The particle's velocity v is computed according to (11). The new velocity depends not only on its previous velocity and position, but also on p_{best} and g_{best} (figure 8). The relative importance of each term of the equation can be changed by modifying the cognitive and social coefficients c_1 and c_2 , as well as the inertia weight w_t which determines the influence of the previous velocity. To enable the algorithm to search the solution space and to make sure that all particles do not get stuck in a local minimum, two random values r_1 and r_2 are also generated between 0 and 1.

$$v_{k+1} = w_t \cdot v_k + \underbrace{c_1 \cdot r_1 \cdot (p_{\text{best}} - x_k)}_{\text{cognitive component}} + \underbrace{c_2 \cdot r_2 \cdot (g_{\text{best}} - x_k)}_{\text{social component}} \quad (11)$$

- The particle's position x is deduced from its previous position and its current velocity:

$$x_{k+1} = x_k + v_{k+1} \quad (12)$$

- Its fitness, a measure of the solution's performance, is computed. This function, to be minimized by the algo-

rithm, represents the total cost of the system:

$$\begin{aligned} \text{Minimize } c_{\text{tot}}(t) &= \sum_{i=0}^{n_{\text{gen}}} c_{\text{gen}}(P_{\text{gen},i}(t)) \\ &+ \sum_{i=0}^{n_s} c_s(P_{s,i}(t)) + \sum_{i=0}^{n_g} c_g(P_{g,i}(t)) \end{aligned} \quad (13)$$

$$\begin{aligned} \text{Subject to } P_{\text{imb}}(t) &= \sum_{i=0}^{n_{\text{gen}}} P_{\text{gen},i}(t) \\ &+ \sum_{i=0}^{n_s} P_{s,i}(t) - \sum_{i=0}^{n_l} P_{l,i}(t) - \sum_{i=0}^{n_g} P_{g,i}(t) = 0 \end{aligned} \quad (14)$$

where $c_{\text{tot}}(t)$ is the total generation cost, n_{gen} , n_s , n_g and n_l are respectively the number of generating units, of storage units, of distribution power lines and of loads. Their respective costs c_X and power outputs P_X use the same indexes. The power balance constraint is taken into account by the fitness function by giving it more weight than the actual solution cost. The total cost computation also use the performance coefficients of the fuel cells.

- The particle's best remembered position is updated.
- The same instructions are executed until the maximum number of iterations or the minimum error criteria is reached.

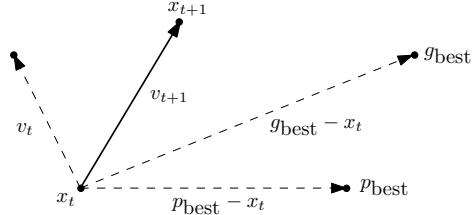


Fig. 8. A PSO algorithm combines each particle's velocity and best solution with the ones of its neighboring group to compute the particle's next position

When a local optimal solution is reached with a PSO algorithm, all particles gather around it, and escaping from this local optimum becomes difficult. To avoid a premature convergence of PSO, we use a novel variant of PSO algorithm, called MPSOM [13], which introduces the following changes:

- The social component of each particle in (11) is computed as a weighted average of all solutions which are better than its own. This allows the particles to prefer following a group of particles rather than a single one. In (15), N_k is a set of solutions which are better than the particle and v_{max} is the maximum velocity.

$$v_{k+1} = w_t \cdot v_k + c_1 \cdot r_1 \cdot (p_{\text{best}} - x_k) + \min \left(v_{\text{max}}, \sum_{i=1}^{|N_k|} \frac{p_{\text{best}} - x_k}{i} \right) \quad (15)$$

- The value of p_{best} is updated according to the Metropolis rule which gives the probability of accepting the position x of a particle. The rule (16) uses a random variable r_3

and a variable T representing a temperature, similarly to temperature in the physical annealing process [14]. This is also to avoid getting stuck in local minima.

$$p = \begin{cases} 1 & \text{if } f(x) \leq f(p_{\text{best}}) \text{ or} \\ & r_3 \cdot \exp\left(\frac{f(x) - f(p_{\text{best}})}{T}\right) < 1 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

- If g_{best} has not been improved after a given number of iterations (here 60), a mutation operator gives the particles the maximum allowed velocity to help them escape a local optimum.

Results described in [13] show that MPSOM has excellent performance in several of the most common tests used in the optimization literature.

E. Fuel cell management algorithm

The control agent also has the capacity to decide when and whether to start or stop fuel cells individually. Such decisions are taken based on forecasts sent by the load agent. By comparing how much power the load will need and how much power the running fuel cells can output at best, the algorithm can decide to start additional fuel cells or to stop the ones that would not be needed anymore. The process takes into account the time required by a fuel cell to start, set at 20 min, as well as the performance coefficients of the fuel cells: the ones with the best coefficients are started first and stopped last.

This capacity is enabled by the multi-agent structure of the system, and can also accommodate connecting or disconnect storage units and power lines similarly to what is possible for fuel cells, and even if the component is previously unknown from the control system. The control agent includes a database which lists all the agents (and the corresponding components) connected to the system and their address. This allows it to acknowledge when an agent connects or disconnects and to communicate with them.

V. SIMULATION RESULTS

In order to test the system we developed, a series of simulations were run to determine how the optimization algorithm performs for this application, how the automatic start and stop functionalities behaves, and finally how storage and power lines can be useful.

A. Optimization tests

In this first category of simulations, we want to observe how the MPSOM algorithm behaves in two cases. The first case corresponds to one round of optimization used to compute the set points to send to the microgrid components when a load and six fuel cells are connected. Each round runs on 1000 iterations before stopping and returning the best solution it could find. The results, pictured in figure 9, show the values of the fitness of each iteration of the algorithm. One can clearly observe that the fitness continuously improves on average (a linear fit was added to show the trend), and that its median value is far better than its mean value. It is also visible that the

algorithm continuously tries to explore other regions than the one where it starts to converge, which can be deduced from the high fitness values obtained throughout the 1000 iterations.

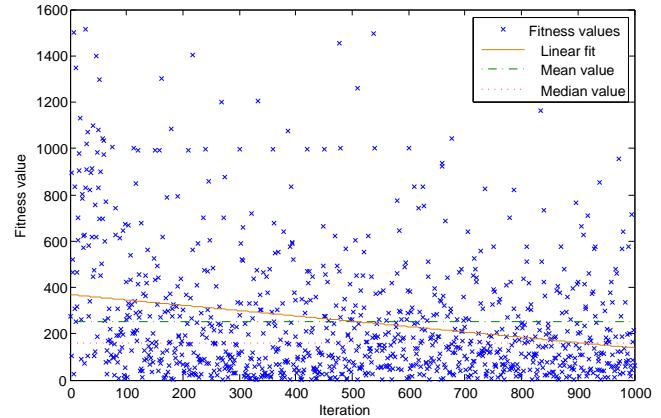


Fig. 9. A first test on the MPSOM algorithm shows that it progressively converges but does not stop looking in the search space for other potentially better solutions. The best result is the one with the lowest fitness.

The second case, in figure 10, is run in a slightly different configuration: a fuel cell has failed and a battery is discharging to provide the necessary power to meet demand. The algorithm needs to find the correct operating point for the battery to meet the constraint on power, which means that the search space is much bigger (see figure 7) and that the fitness can increase to higher values than before. However, despite those changes, the behavior of MPSOM is similar to the previous test case and manages to quickly find the region where the optimum is located.

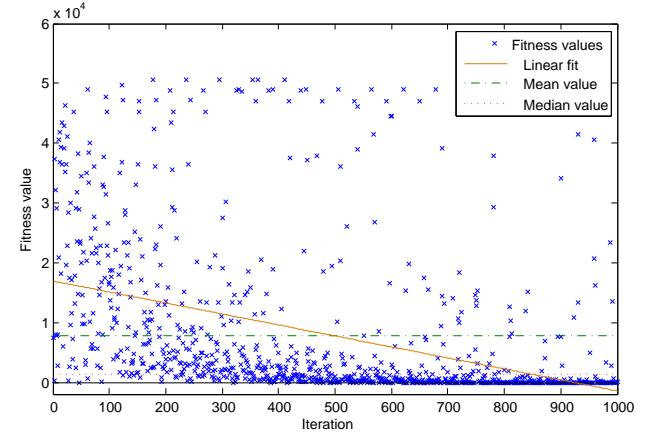


Fig. 10. This second test is run as a fuel cell underwent a failure and a storage unit discharges to match demand. The search space is much bigger in this case, due to the use of the battery, but the algorithm also manages to converge.

B. Cost efficiency tests

In order to test if the method presented here is really economically efficient, two simulations are run and compared. A first test serves as a reference case: enough fuel cells are

started to meet demand peaks, no storage is used and the fuel cells are supposed to be identical (the performance coefficients are disregarded). Automatically starting and stopping fuel cells is also disabled. The second test's objectives are to observe the behavior of the system when more functionalities of the algorithm are used: the fuel cells are non-identical and have different performance coefficients, and can be started and stopped by the algorithm.

To run those tests, a time interval of 60 s between each computing step is chosen. This value is considered as high enough to be used with an energy management algorithm and low enough to ensure that generation meets demand on a regular basis. The load profile was adapted from Southern California Edison's load profiles [15]. This profile, shown in figure 12, represents the consumption measured on a substation in an industrial area over a period of 4 days. It starts with two weekend days, when the consumption is relatively low and ends with two weekdays when demand increases a lot during daytime.

The first test results are displayed in figure 11. The total cost after going through the whole demand profile is of about € 3500, only for hydrogen costs set to € 2.02 per kg. The second test (figure 12) results in a total cost lowered to € 2600 for the same load profile, meaning that non negligible savings can be achieved. This difference results from the facts that only the required number of fuel cells are started to match demand and that the least performing fuel cells are less used. For example, we can clearly see that at least one of the fuel cells is only started during the two biggest demand peaks.

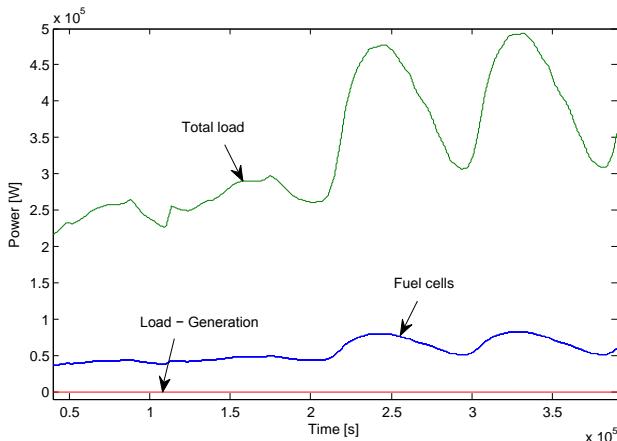


Fig. 11. The first test serves as a reference scenario, where all the energy management functionalities of the algorithm are disabled. The load is shared equally between all fuel cells.

C. Tests on the use of storage and power lines

The last category of tests aims at showing how storage and grid can be used in this paradigm. Figure 13 shows what happens when a fuel cell fails and a battery quickly responds to this sudden and unforeseen event. This allows the system to keep the balance between generation and demand. Additionally to storage, the algorithm also uses its ability to

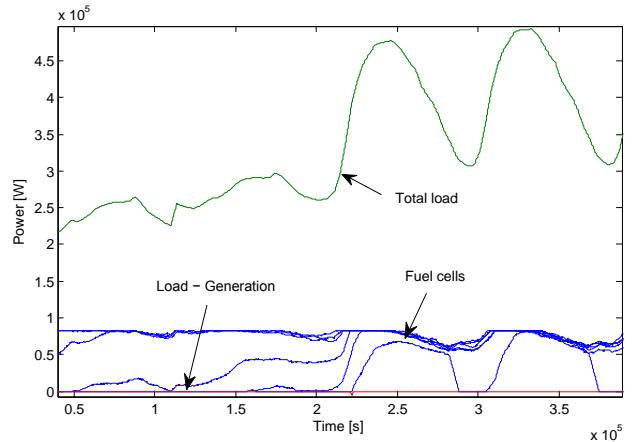


Fig. 12. The second test uses the same profile as the first one but results in a much lower total cost thanks to the functionalities offered by the algorithm

start other fuel cells which were not used at that time. This allows the battery to simply make the transition until the fuel cells are started and avoids storage to lose too much capacity.

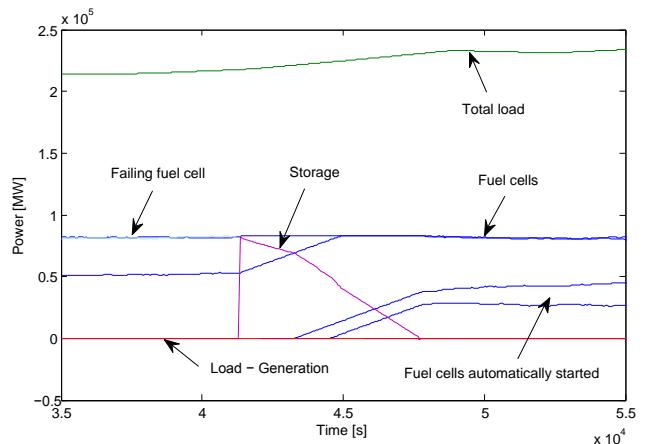


Fig. 13. This simulation shows that storage units such as batteries can help the system face unforeseen conditions such as a generator failure. Two additional fuel cells are also quickly started so that the battery does not become completely discharged.

In figure 14, a similar test is done but with the main grid, from which energy can be bought and sold, instead of a battery. The behavior of the grid agent is very close to the one of the battery, except that importing energy from the grid costs money. However, as making demand and generation match is considered as more important in the fitness calculation, this cost has almost not impact in this case. Deciding between using storage or grid for this situation depends mostly on the cost of storage, which is still quite high and on the availability of a power line.

VI. CONCLUSION AND FUTURE WORK

This article has shown that a fuel cell array can be efficiently controlled by a multi-agent-based energy management system to minimize its costs while meeting demand. The application

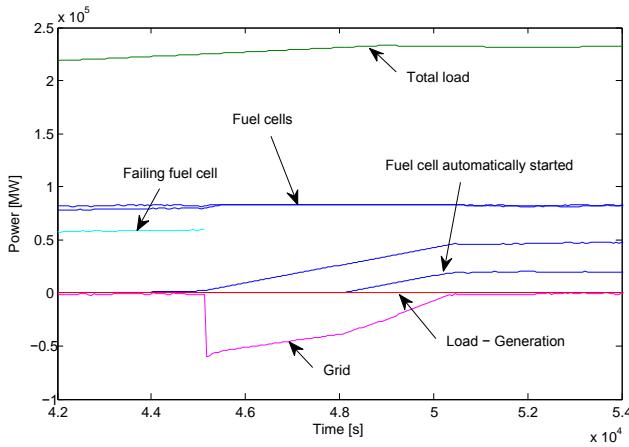


Fig. 14. In this case, power is bought from the grid to achieve results similar to the ones obtained with the battery. Power bought from the grid is counted as negative.

of the MPSOM algorithm to this problem shows a good performance without requiring special adaptations. The ability to automatically start and stop fuel cells, as well as the capacity to control each of them according to their own performance are promising tools that will be further developed in the future and can help achieve significant cost savings. The multi-agent architecture of the control system also results in a high degree of flexibility, and can accommodate sudden changes in the system structure. Although the use of fuel cell arrays is still very seldom, especially as primary power sources, similar principles could be applied to other types of controllable energy sources. Storage and grid can also provide additional degrees of freedom, especially when generators cannot follow demand or fail.

Current and future work follow two main directions:

- Using more precise component and grid models and adding more types of sources and loads.
- Developing smarter controls for storage and grid, for decision-making regarding starting and stopping generators (this should be limited to prevent a premature aging of the fuel cells) and for allowing demand-side management.
- Modeling the corresponding grid in a power systems simulation software to analyze the behavior of the system

(on voltage, etc.) with this energy management algorithm.

- Analyzing the performance of the algorithm in terms of speed and of the MAS in terms of bandwidth needs.

REFERENCES

- [1] Bloom Energy. (2011) ES-5000 Energy Server. [Online]. Available: <http://www.bloomenergy.com/products/data-sheet/>
- [2] S. Fairfax and N. Dowling, "Operation and economics of a behind-the-meter fuel cell power plant array," in *Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES)*, 2010 IEEE Conference on, 2010, pp. 278–284.
- [3] J. Amphlett, R. Mann, B. Peppley, P. Roberge, and A. Rodrigues, "A model predicting transient responses of proton exchange membrane fuel cells," *Journal of Power Sources*, vol. 61, no. 1-2, pp. 183 – 188, 1996.
- [4] J. Ferber, *Multi-Agent Systems: An Introduction to Artificial Intelligence*. Addison-Wesley, 1999.
- [5] A. S. Rao and M. P. Georgeff, "Modeling rational agents within a BDI-architecture," in *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, J. Allen, R. Fikes, and E. Sandewall, Eds. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991, pp. 473–484. [Online]. Available: <http://jmvidal.cse.sc.edu/library/rao91a.pdf>
- [6] R. Roche, B. Blunier, A. Miraoui, V. Hilaire, and A. Koukam, "Multi-agent systems for grid energy management: A short review," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, nov. 2010, pp. 3341–3346.
- [7] S. McArthur, E. Davidson, V. Catterton, A. Dimeas, N. Hatzigargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications - Part I: Concepts, Approaches, and Technical Challenges," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1743–1752, nov. 2007.
- [8] ———, "Multi-Agent Systems for Power Engineering Applications - Part II: Technologies, Standards, and Tools for Building Multi-agent Systems," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1753–1759, nov. 2007.
- [9] J. Lagorse, D. Paire, and A. Miraoui, "A multi-agent system for energy management of distributed power sources," *Renewable Energy*, vol. 35, no. 1, pp. 174–182, 2010.
- [10] F. Bellifemine, G. Caire, T. Trucco, and G. Rimassa, "Jade programmer's guide," Tech. Rep., april 2010. [Online]. Available: <http://jade.tilab.com/doc/programmersguide.pdf>
- [11] Foundation for Intelligent Physical Agents (FIPA). (2011). [Online]. Available: www.fipa.org
- [12] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. Morgan Kaufmann Academic Press, 2001.
- [13] L. Idoumghar, M. Idrissi-Aouad, M. Melkemi, and R. Schott, "Metropolis particle swarm optimization algorithm with mutation operator for global optimization problems," in *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, vol. 1, oct. 2010, pp. 35–42.
- [14] M. Locatelli, "Simulated annealing algorithms for continuous global optimization," *Handbook of global optimization*, vol. 2, pp. 179–229, 2002.
- [15] Southern California Edison. (2011) SCE load profiles. [Online]. Available: <http://www.sce.com/AboutSCE/Regulatory/loadprofiles/2011loadprofiles.htm>

Imperialist Competitive Algorithm for Dynamic Optimization of Economic Dispatch in Power Systems

Robin Roche¹, Lhassane Idoumghar², Benjamin Blunier¹, and Abdellatif Miraoui¹

¹ Université de Technologie de Belfort-Montbéliard,
Laboratoire Systèmes et Transports, 90010 Belfort, France,

robin.roche@utbm.fr

² Université de Haute-Alsace,
LMIA / INRIA Grand Est, 68093 Mulhouse, France,
lhassane.idoumghar@uha.fr

Abstract. As energy costs are expected to keep rising in the coming years, mostly due to a growing worldwide demand, optimizing power generation is of crucial importance for utilities. Economic power dispatch is a tool commonly used by electric power plant operators to optimize the use of generation units. Optimization algorithms are at the center of such techniques and several different types of algorithms, such as genetic or particle swarm algorithms, have been proposed in the literature. This paper proposes the use of a new metaheuristic called imperialist competitive algorithm (ICA) for solving the economic dispatch problem. The algorithm performance is compared with the ones of other common algorithms. The accuracy and speed of the algorithm are especially studied. Results are obtained through several simulations on power plants and microgrids in which variable numbers of generators, storage units, loads and grid import/export lines are connected.

Keywords: metaheuristic, imperialist competitive algorithm, dynamic optimization, economic dispatch, microgrid

1 Introduction

With fossil resources becoming harder and harder to extract and worldwide demand continuously increasing, energy costs are expected to rise significantly in the coming years. Therefore, optimizing the use of these resources is of crucial importance in order to minimize costs. Such considerations are essential to power plant and transmission grid operators, for which optimizing operating costs can result in significant savings and profit. Economic power dispatch enables such an optimization and aims at determining the most cost-efficient and reliable operation of power systems, such as power plants. This objective is achieved by optimally dispatching available generation resources to supply the load connected to the system.

Optimization algorithms play an important role in economic dispatch, as they are the central tool used to obtain the optimal dispatch. Over the years, numerous algorithms have been used but imperialist competitive algorithms, a new type of metaheuristic based on imperialistic competition, are still largely unexplored for this problem. Moreover, very few were utilized in a dynamic optimization context.

The following studies whether this algorithm could and should be more widely used for this application. It describes the economic dispatch optimization problem, the imperialist competitive algorithm we used, how we tested it on mathematical functions and compared it with other common algorithms and finally how it was tested on two power systems for solving the dynamic economic dispatch problem.

2 Economic Power Dispatch Problem

2.1 Economic Dispatch Concept

The US Energy Policy Act [10] defines economic dispatch (ED) as “the operation of generation facilities to produce energy at the lowest cost to reliably serve consumers, recognizing any operational limits of generation and transmission facilities.” In short, ED aims at optimally dispatching power generation between available generation units to meet demand. Its objectives are usually to minimize fuel utilization and sometimes also greenhouse gases emissions, compared to what less efficient generation sources would result in. Finding the best trade-off between costs, environmental impact and reliability is thus the main challenge of ED. In order to reduce operation costs, ED has been used in various forms for years, especially by transmission system and power plant operators.

Two types of ED can be considered: economic dispatch for the current day, and for future days.

- The first one, sometimes referred to as load following, consists in dispatching power for the current day, by monitoring load, generation and power imports/exports, while ensuring a balance between load and supply. The stability of the frequency of the grid (50 or 60 Hz) is the consequence of this balance, and is required by most loads which use it as a reference.
- The second one corresponds to dispatch for the following day, or several days after. Performed by the generation group or an independent market operator, it mainly consists in scheduling generators for each hour of the next day’s dispatch, based on load forecasts. The units to use are selected based on their characteristics, costs and maintenance requirements.

The following will focus on the first type of dispatch, for the current day. Generators, such as gas turbines or fuel cells, will not be scheduled but will be sent set points their own control system should take into account immediately, while ensuring the reliability of the system. However, algorithms similar to the ones described in the following can be used for scheduling and committing generators. It should also be mentioned that determining if an algorithm behaves well

for ED is a necessary task, as the consequences of a bad ED can have a significant impact on costs for the operator and the consumer and cause instabilities or even blackouts.

2.2 Objective, Constraints and Hypotheses

To achieve an optimal power dispatch from an economic point of view, an objective function needs to be defined. This function (1) corresponds to the total cost of generation, storage, and power imports and exports. In order to maintain the grid frequency stable and the system reliability, constraint (2), requiring a balance between generation and supply, must be met.

$$\text{Minimize } c_{\text{tot}}(t) = \sum_{i=0}^{n_{\text{gen}}} c_{\text{gen}}(P_{\text{gen},i}(t)) + \sum_{i=0}^{n_s} c_s(P_{s,i}(t)) + \sum_{i=0}^{n_g} c_g(P_{g,i}(t)) \quad (1)$$

$$\text{Subject to } P_{\text{imb}}(t) = \sum_{i=0}^{n_{\text{gen}}} P_{\text{gen},i}(t) + \sum_{i=0}^{n_s} P_{s,i}(t) - \sum_{i=0}^{n_l} P_{l,i}(t) - \sum_{i=0}^{n_g} P_{g,i}(t) = 0 \quad (2)$$

where $c_{\text{tot}}(t)$ is the total generation cost, n_{gen} , n_s , n_g and n_l are respectively the number of generating units, of storage units, of grid import/export lines and of loads. Their respective costs c_X and power outputs P_X use the same indexes.

In the following, constraint (2) is taken into account by the algorithm through its objective function. The actual fitness or objective function $f(t)$ (3) is a combination of the total cost defined in (1) and of the imbalance between load and supply. A coefficient α is set according to the ratio between the magnitude of the power system (W, kW, MW, etc.) and the estimated costs to give more importance to keeping the imbalance as low as possible while minimizing costs.

$$\text{Minimize } f(t) = c_{\text{tot}}(t) + \alpha \cdot P_{\text{imb}}(t) \quad (3)$$

Additional constraints are also to be respected, notably for the optimization bounds. These bounds reflect the characteristics and dynamics of the controlled units. For example, a generating unit i such as a gas turbine can only operate within its operating range (4), and its power output variation is limited by ramp-up and ramp-down limits R_u and R_d (5). In other words, the bounds at time $t+1$ depend on the ones at t because of the dynamic of the units and are thus updated at every call of the algorithm by an expert system.

$$P_{i,\min} \leq P_{i,t} \leq P_{i,\max} \quad (4)$$

$$P_{i,t} - R_{i,d} \leq P_{i,t+1} \leq P_{i,t} + R_{i,u} \quad (5)$$

In addition to these objectives and constraints, the following simplifying assumptions are made, as the primary focus of this paper is the performance of the algorithm:

- Valve-point effects, reactive power, line losses and emissions (e.g., CO₂ or NO_x) are not considered.

- All voltage magnitudes are assumed to be nominal. Bus and node capacities are assumed to be sufficient.
- Scheduling, starting and stopping generators is not achieved by the algorithm.

2.3 Problem Characteristics

The ED problem, as we will treat it, can be considered as a dynamic, non-linear, constrained, optimal control problem. It is at first dynamic because the cost function and the constraints change or can change at every call of the algorithm. The controlled power system is indeed itself a dynamic system, which evolves over time. A gas or wind turbine can for example be stopped for maintenance. Due to this dynamic nature, the algorithm is run at a fixed frequency, which can be as low as a few seconds. To a certain extent, it can also be considered as an online optimization problem, as the optimal control is performed in almost real-time. This is enabled by the small scale of the considered systems.

The variables to optimize are control set points corresponding to powers that are defined in continuous spaces, themselves defined by the energy management policy of the operator and by an expert system based on models. Algorithms able to operate in real-valued search spaces are thus required, as well as algorithms which do not require precise information on the search space or on the objective function. The reason for this is that the algorithm should be able to adapt to as many systems as possible without requiring much effort. Stochastic or randomized search methods are ideal for such constraints.

2.4 Optimization Algorithms for this Problem

The ED problem is a classical optimization problem in power systems and has been solved using numerous optimization algorithms:

- Linear programming [6, 5], for which the objective function and the constraints must be linear. As we do not want to have any hypothesis on the shape of the functions, those algorithms are not selected. However, they are commonly used in the industry due to their speed and relative simplicity, but with constraint relaxation methods (such as Lagrange multipliers) and cost functions approximated by quadratic functions.
- Combinatorial algorithms, such as dynamic programming [11], can include integer variables and are based on the idea of splitting the problem into subproblems. Although these algorithms can be used for scheduling sources, their characteristics do not fit the needs of the current problem.
- Metaheuristics, such as genetic algorithms [12], particle swarm optimization [3] or simulated annealing. This last category of algorithms makes few or no assumption about the problem (e.g., the objective function does not need to be differentiable) but do not guarantee an optimal solution is ever found as they rely on random variables. As we do not know how the objective function looks like and evolves over time, possibly with discontinuities, these algorithms are selected.

Although the variety of these algorithms is quite large, the literature contains few examples of imperialist competitive algorithms applied to the solving of problems in power systems [2], and none for the ED problem specifically.

3 Imperialist Competitive Algorithm

The imperialist competitive algorithm (ICA) is a new evolutionary optimization approach introduced in 2007 by E. Atashpaz-Gargari [1]. It is inspired by the imperialistic competition processes of human societies. The algorithm can be seen as a social counterpart of genetic algorithms. Several of its steps are indeed similar: countries can undergo revolutions as chromosomes can mutate, for example.

This algorithm uses a precise terminology, in which a solution is called a country. There are two types of countries: imperialist countries, and colonies, which depend on these imperialists. An imperialist and its countries form a group of countries called empire.

Algorithm 1 Imperialist Competitive Algorithm

```

1: Initialize the countries and form the empires
2: while the stop condition is not satisfied do
3:   for all empires do
4:     Move the colonies toward the imperialist (assimilation)
5:     Make some colonies undergo a revolution
6:     if a colony is more powerful than the imperialist then
7:       The colony becomes the imperialist and vice versa (overthrow)
8:     end if
9:   end for
10:  if two empires are too close then
11:    Merge them (unification)
12:  end if
13:  Make imperialistic competition occur
14:  if there is an empire with no colonies then
15:    Eliminate this empire
16:  end if
17: end while

```

ICA works as illustrated in Algorithm 1, where the following processes are used:

- Initialization and empire formation: Like other evolutionary algorithms, ICA starts with an initial population of solutions called countries, of size N_{pop} . Among them, the N_{imp} best countries (the most powerful) are selected to be imperialists. The remaining N_{col} countries form the colonies of these imperialists. The n initial empires are formed by dividing the colonies among impe-

rialists according to their normalized power P derived from their cost c (6).

$$P_n = \left| \frac{c_n - \max_i c_i}{\sum_{i=1}^{N_{\text{imp}}} (c_n - \max_i c_i)} \right| \quad (6)$$

The number of colonies $N_{\text{col},n}$ attached to empire n is computed according to (7).

$$N_{\text{col},n} = \text{round}(P_n \cdot N_{\text{col}}) \quad (7)$$

- Assimilation: Imperialist countries attract colonies to them using the assimilating policy illustrated in Fig. 1. To update its position x , each colony moves toward its imperialist by updating its position using (8).

$$x_{t+1} = x_t + \beta \cdot \gamma \cdot r \cdot d \quad (8)$$

where $\beta > 1$ causes the colonies to get closer to the imperialist, $\gamma < 1$ corresponds to an assimilation coefficient, r is a random number chosen from the uniform distribution $\mathcal{U}(-\theta, \theta)$, θ adjusts the deviation from the original direction and enables searching around the imperialist and d is the distance between the colony and the imperialist.

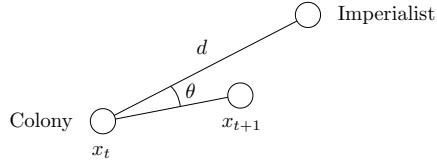


Fig. 1. Movement of a colony toward its imperialist

- Revolution: The revolution process introduces sudden random changes in the position of some countries. It plays the same role as the mutation operator in a genetic algorithm.
- Overthrow: After assimilation and revolution, a colony might reach a better position than the imperialist of the empire. In this case, the colony can become the imperialist and vice versa.
- Unification: If two empires are too close to each other, they can unite and become a single empire, with the sum of the colonies of the two initial empires.
- Imperialistic competition: Each empire tries to take possession of colonies of other empires and control them. This imperialistic competition is modeled by selecting the weakest colonies of the weakest empire and giving them to the empire that has the highest likelihood to possess them.

The total power $P_{\text{tot},n}$ of each empire n is defined by the power of its imperialist plus its average colonies' power, as defined in (9) where $\zeta \ll 1$, I refers to the empire's imperialist and C to its colonies.

$$P_{\text{tot},n} = P(I_n) + \zeta \cdot \text{mean}(P(C_n)) \quad (9)$$

The likelihood p_n , called possession probability, is then derived from each empire's power (10).

$$p_n = \left| \frac{c_{\text{tot,norm},n}}{\sum_{i=1}^{N_{\text{imp}}} c_{\text{tot,norm},i}} \right| \quad (10)$$

where $c_{\text{tot,norm},n} = c_{\text{tot},n} - \max_i c_{\text{tot},i}$ is the total normalized cost of empire n and $c_{\text{tot},n}$ its total cost.

In order to divide the colonies among empires based on their possession probability, a vector A is built (11), where P_n is the power of empire n and r_n a random value between 0 and 1. The selected colonies are then assigned to the empire whose relevant index in A is maximum.

$$A = [P_1 - r_1, P_2 - r_2, \dots, P_{N_{\text{imp}}} - r_{N_{\text{imp}}}] \quad (11)$$

4 Performance on Mathematical Problems

Before testing the algorithm on the ED problem, its performance is compared to other evolutionary algorithms. These algorithms are:

- Metropolis Particle Swarm Optimization Algorithm with Mutation Operator (MPSOM) [4]: This hybrid PSO variant uses the Metropolis rule and a mutation operator to avoid local minima.
- Differential Evolution (DE) [8]: This algorithm combines the positions of solutions, called agents, to move them in the search space. Only the moves that lead to an improvement are accepted, others are discarded.
- Imperialist Competitive Algorithm (ICA), which was just presented.

The parameters of the ICA were empirically determined by running iterative trials using the mathematical functions described in Table 2, and starting with the parameters given by the authors in [1]. The tuned parameters are summarized in Table 1.

Several benchmark functions [9] described in Table 2 were used to test the algorithms performance. As the focus is primarily on the ED problem, the number of functions is limited to four. These functions provide a good start for testing the credibility of an optimization algorithm. Each of these functions has many local optima in its solution space. The amount of local optima increases with their dimension, which was set to 20 as in [7]. For each algorithm, the maximum number of function evaluations is 300,000. A total of 30 runs for each algorithm was conducted and the average fitnesses of the best solutions were recorded.

The mean solutions and the corresponding standard deviations obtained for the algorithms are listed in Table 3. MPSOM obtains the best results for the first two functions, and DE for the third, Rosenbrock. ICA performs best on Ackley, with slightly better results than DE. A finer tuning of ICA parameters

Table 1. Parameter settings for the ICA approach

Parameter	Variable	Value
Number of initial solutions	N_{pop}	60
Number of initial imperialists	N_{imp}	6
Maximum number of iterations	N_d	5,000
Revolution rate	R_r	0.1
Assimilation coefficient	β	2
Assimilation angle coefficient	θ	$\frac{\pi}{6}$
ζ coefficient	ζ	0.02
Uniting threshold	U_t	0.02

Table 2. Standard benchmark functions adopted in this work

Function	Problem	Range
Sphere	$\sum_{i=1}^n x_i^2$	[-100;100]
Rastrigin	$\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[-5.12;5.12]
Rosenbrock	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[-2.048;2.048]
Ackley	$20 + e - 20 e^{-0.2 (\frac{1}{n} \sum_{i=1}^n x_i^2)^{\frac{1}{2}}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)}$	[-30.0;30.0]

Table 3. Comparison of the solutions obtained by the four selected metaheuristics. No lower threshold limit is set for the results.

Function	MPSOM	DE	ICA
Sphere	1.17×10^{-114}	7.145×10^{-40}	8.458×10^{-12}
	$\pm 1.17 \times 10^{-114}$	$\pm 1.390 \times 10^{-39}$	$\pm 2.532 \times 10^{-11}$
Rastrigin	0	4.551	2.251×10^{-06}
	± 0	± 0.919	$\pm 3.957 \times 10^{-06}$
Rosenbrock	1.44×10^{-02}	1.880×10^{-11}	0.3209
	$\pm 1.63 \times 10^{-02}$	$\pm 1.211 \times 10^{-11}$	± 0.4014
Ackley	1.16×10^{-10}	3.946×10^{-15}	8.025×10^{-16}
	$\pm 2.42 \times 10^{-11}$	$\pm 1.119 \times 10^{-16}$	$\pm 8.365 \times 10^{-15}$

and hybridizing it with another algorithm would probably strongly improve its results on such functions. The following examines how the algorithm performs on the ED problem.

5 Simulations for the Economic Dispatch Problem

Two different grid configurations are used to test the performance of the algorithm, one is a microgrid where the stability of the algorithm will be tested and the other a fuel cell power plant where the optimization will focus on the total cost of the simulation. Both tests run over a period of four to five days and are based on real load profiles³. The optimization process is run every 60 s. Coefficient α from (3) is set to 10^5 . The tests were run several times and returned very similar results.

5.1 Microgrid Test

The first test corresponds to a microgrid with two identical 83 kW fuel cells, photovoltaic panels with a rated peak power of 600 kW, two 500 kW wind turbines, a 1 MW import/export power line and a group of loads corresponding to the consumption of a residential area. The proportion of renewable energy sources in the microgrid corresponds to a very high penetration rate, which often implies control difficulties due to their intermittency. Simple models are used to extract the sources' power output from input values such as solar irradiation and wind speed⁴, as well as to extract the hydrogen consumption of the fuel cells.

The expert system, which reflects the merit order chosen by the operator, determines the optimization bounds for each element. Renewable and non-controllable energy sources are used in priority and controllable generation units such as fuel cells are only allowed to provide the missing power to meet demand. Importing or exporting energy to the main grid is only allowed when the other sources are not sufficient (a similar rule could be used for storage).

The cost functions of the fuel cells reflect their fuel consumption, and generation costs from renewable energy sources are considered as equal to zero. The cost for importing a given amount of energy from the grid is lower than the price at which the same amount is bought by the grid, according to the principles of feed-in tariffs.

The test is run with three algorithms: ICA, DE and MPSOM. Results are shown in Fig. 2 for ICA and in Table 4 for all algorithms. They show that the algorithms perform well regarding the accuracy of constraint verification, i.e., the balance between load and supply (not displayed in Fig. 2 for the sake of clarity) is maintained. DE is the fastest but also the least efficient for minimizing total costs. MPSOM performs well, as well as ICA, except for its duration.

5.2 Fuel Cell Power Plant Test

The second configuration is a power plant based on a fuel cell array. A group of six 83 kW fuel cells generate power and supply a local load assimilable to an industrial and commercial area.

³ Adapted from Southern California Edison's load profiles at:
<http://www.sce.com/AboutSCE/Regulatory/loadprofiles>

⁴ Extracted from: <http://www.unige.ch/cuepe/html/donnees-csv.php>

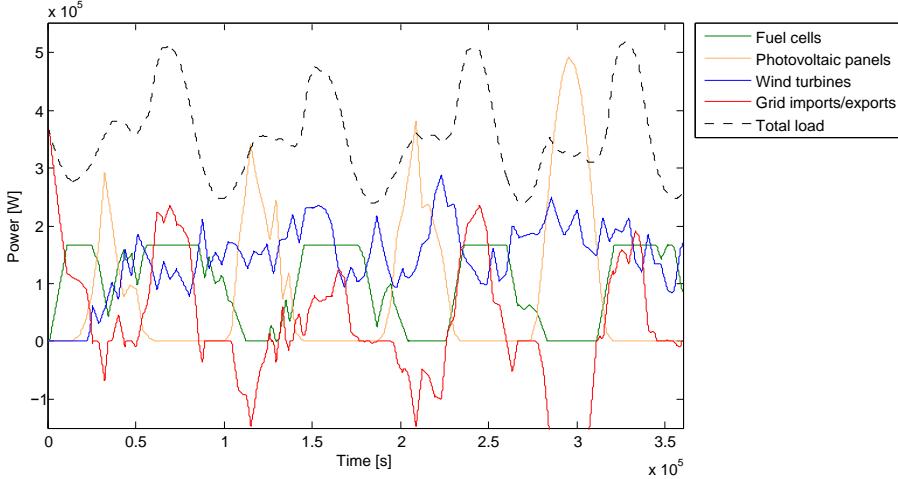


Fig. 2. Results of the microgrid simulation with ICA. Although intermittent sources induce large variations in generation, the use of fuel cells and of grid imports/exports, and the corresponding costs, are minimized by the algorithm while verifying the power balance constraint.

Table 4. Microgrid test results comparison

Algorithm	Unit	MPSOM	DE	ICA
Total cost	€	1,075	1,155	1,088
Mean imbalance	W	-0.0175	0.1344	0.1314
Mean duration	ms	8.126	4.075	257.0

The cost functions of the fuel cells differ from one fuel cell to the other. The aging and wearing of the units is taken into account by adding a multiplier coefficient η (12), which degrades or improves the cost of the fuel cells according to their age and how they are used and maintained. Coefficients between 0.9 and 1.2 are selected (see Fig. 3's legend – a high coefficient corresponds to an old and more expensive fuel cell), and are used to determine which algorithm behaves best for this kind of optimization.

$$c_{fc}(P_{fc}) = \eta \cdot m((P_{fc})) \quad (12)$$

where c_{fc} is the cost of the fuel cell for set point P_{fc} , and m the fuel cost function.

Table 5 summarizes the results of the tests shown in Fig. 3 for ICA. The same test profile is run with the three algorithms used in the previous test and a reference algorithm called equal dispatch (EqD), which simply consists in dividing the total load by the number of operating fuel cells without any optimization process.

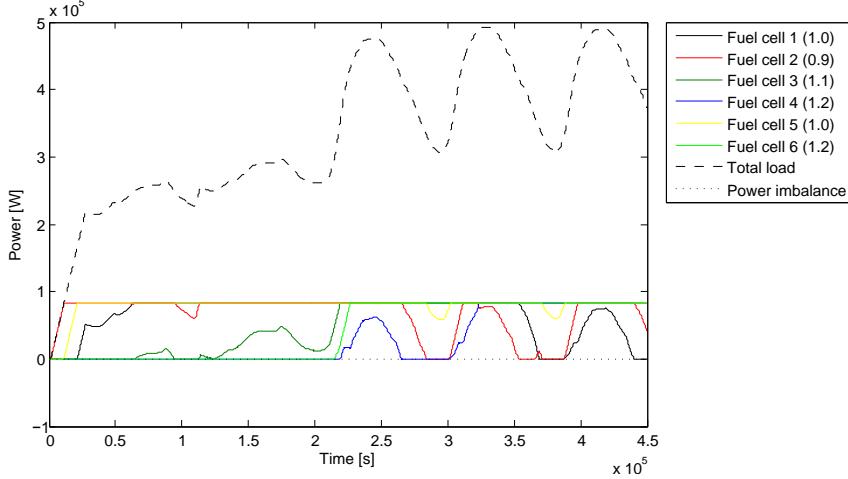


Fig. 3. Results of the power plant simulation with ICA. By properly selecting how much to use each fuel cell, the algorithm manages to achieve the best total cost. Older and less efficient fuel cells indeed tend to be less used by the algorithm to meet demand.

Table 5. Fuel cell power plant test results comparison

Algorithm	Unit	EqD	MPSOM	DE	ICA
Total cost	€	4,097	2,165	2,160	2,119
Mean imbalance	W	$\simeq 0$	-0.099	-2.286	-2.888
Mean duration	ms	< 1	11.82	8.391	454.8

Results show that ICA gives the best results in terms of total cost, as opposed to EqD. However, as with the previous test, it is also slower than the other algorithms. DE is particularly fast but is also rather imprecise, although this imbalance only represents $10^{-4}\%$ of the maximum total load. It should also be noted that the imbalance for ICA is almost always close to zero except for a few minutes around 4×10^5 s, where the imbalance reaches 500 W.

6 Conclusion and Future Work

An application to dynamic ED of a new metaheuristic based on imperialist competition, the ICA, was described in this article. It showed that ICA provides good results for the ED problem, which was the aim of this work. It helps minimize costs even more than the other tested algorithms, while maintaining a good accuracy regarding the constraint. However, it is slower and performs less well on mathematical functions, illustrating the famous “no free lunch theorem”. It therefore needs to be improved by further tuning its parameters and by hybridiz-

ing it with another optimization algorithm, similarly to what was achieved with MPSOM. Future work will also focus on speeding up the algorithm.

As the objective of this article was to focus on the performance of the algorithm for the dynamic ED problem, several simplifying assumptions were made. However, in the future, emissions of classical energy sources will be taken into account to reflect current environmental concerns. Deciding whether generation sources should be started or stopped, as well as demand-side management, will also be integrated in the algorithm, which will enable next-day scheduling.

References

1. Atashpaz-Gargari, E., Lucas, C.: Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In: IEEE Congress on Evolutionary Computation. pp. 4661–4667 (2007)
2. Duki, E.A., Mansoorkhani, H.R.A., Soroudi, A., Ehsan, M.: A discrete imperialist competition algorithm for transmission expansion planning. In: 25th International Power System Conference (2010)
3. El-Gallad, A., El-Hawary, M., Sallam, A., Kalas, A.: Particle swarm optimizer for constrained economic dispatch with prohibited operating zones. In: Canadian Conference on Electrical and Computer Engineering. vol. 1, pp. 78–81 (2002)
4. Idoumghar, L., Idrissi-Aouad, M., Melkemi, M., Schott, R.: Metropolis particle swarm optimization algorithm with mutation operator for global optimization problems. In: 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI). vol. 1, pp. 35–42 (oct 2010)
5. Irving, M., Sterling, M.: Economic dispatch of active power with constraint relaxation. IEE Proceedings C Generation, Transmission and Distribution 130(4), 172–177 (1983)
6. Nabona, N., Freris, L.: Optimisation of economic dispatch through quadratic and linear programming. Proceedings of the Institution of Electrical Engineers 120(5), 574–580 (1973)
7. Pant, M., Thangaraj, R., Abraham, A.: Particle swarm based meta-heuristics for function optimization and engineering applications. In: 7th Conf. Computer Information Systems and Industrial Management Applications. vol. 7, pp. 84–90. IEEE Computer Society (2008)
8. Storn, R., Price, K.: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11(4), 341–359 (1997)
9. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Tech. Rep. 2005005, Nanyang Technological University, Singapore and IIT Kanpur, India (2005)
10. United States Department of Energy: Economic dispatch of electric generation capacity (2007), a report to Congress and the States pursuant to sections 1234 and 1832 of the energy policy act of 2005
11. Waight, J., Albuyeh, F., Bose, A.: Scheduling of generation and reserve margin using dynamic and linear programming. IEEE Transactions on Power Apparatus and Systems PAS-100(5), 2226–2230 (1981)
12. Walters, D., Sheble, G.: Genetic algorithm solution of economic dispatch with valve point loading. IEEE Transactions on Power Systems 8(3), 1325–1332 (1993)

Research Article

Hybrid PSO-SA Type Algorithms for Multimodal Function Optimization and Reducing Energy Consumption in Embedded Systems

Lhassane Idoumghar,^{1,2} Mahmoud Melkemi,² René Schott,³ and Maha Idrissi Aouad¹

¹INRIA Nancy—Grand Est/LORIA, 615 Rue du Jardin Botanique, 54600 Villers-Lès-Nancy, France

²LMIA—MAGE, Université de Haute-Alsace, 4 Rue des Frères Lumière, 68093 Mulhouse, France

³IECN—LORIA, Nancy-Université, Université Henri Poincaré, 54506 Vandoeuvre-Lès-Nancy, France

Correspondence should be addressed to Lhassane Idoumghar, lhassane.idoumghar@uha.fr

Received 31 December 2010; Revised 22 March 2011; Accepted 11 April 2011

Academic Editor: Chuan-Kang Ting

Copyright © 2011 Lhassane Idoumghar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The paper presents a novel hybrid evolutionary algorithm that combines Particle Swarm Optimization (PSO) and Simulated Annealing (SA) algorithms. When a local optimal solution is reached with PSO, all particles gather around it, and escaping from this local optima becomes difficult. To avoid premature convergence of PSO, we present a new hybrid evolutionary algorithm, called HPSO-SA, based on the idea that PSO ensures fast convergence, while SA brings the search out of local optima because of its strong local-search ability. The proposed HPSO-SA algorithm is validated on ten standard benchmark multimodal functions for which we obtained significant improvements. The results are compared with those obtained by existing hybrid PSO-SA algorithms. In this paper, we provide also two versions of HPSO-SA (sequential and distributed) for minimizing the energy consumption in embedded systems memories. The two versions, of HPSO-SA, reduce the energy consumption in memories from 76% up to 98% as compared to Tabu Search (TS). Moreover, the distributed version of HPSO-SA provides execution time saving of about 73% up to 84% on a cluster of 4 PCs.

1. Introduction

Several optimization algorithms have been developed over last few decades for solving real-world optimization problems. Among them, we have many heuristics like Simulated Annealing (SA) [1] and optimization algorithms that make use of social or evolutionary behaviors like Particle Swarm Optimization (PSO) [2, 3]. SA and PSO are quite popular heuristics for solving complex optimization problems, but they have some strengths and limitations.

Particle Swarm Optimization (PSO) is based on the social behavior of individuals living together in groups. Each individual tries to improve itself by observing other group members and imitating the better ones. This way, the group members are performing an optimization procedure which is described in [3]. The performance of the algorithm depends on how the particles (i.e., potential solutions to

an optimization problem) move in the search space, given that the velocity is updated iteratively. Large research body is therefore devoted to the analysis and proposal of different motion rules (see [4–6] for recent accounts of PSO research). To avoid premature convergence of PSO, we combine it with SA: PSO contributes to the hybrid approach in a way to ensure that the search converges faster, while SA makes the search jump out of local optima due to its strong local-search ability. In this paper, we present a hybrid optimization algorithm, called HPSO-SA, which exploits intuitively the positive features of PSO and SA. We also validate HPSO-SA using ten benchmark functions given in [7] and compare the results with classical PSO, ATREPSO, QIPSO, and GMPSO algorithms described in [2], TL-PSO [8], PSO-SA [9], SAPSO and SUPER-SAPSO presented in [10]. We provide also two versions of HPSO-SA (sequential and distributed) for minimizing the energy consumption in

embedded systems memories. The two versions, of HPSO-SA, reduce the energy consumption in memories from 76% up to 98% as compared to Tabu Search (TS). Moreover, the distributed version of HPSO-SA provides execution time saving of about 73% up to 84% on a cluster of 4 PCs.

The rest of the paper is organized as follows. Section 2 introduces, briefly, PSO and SA algorithms. Section 3 is devoted to detailed description of HPSO-SA. In Section 4, benchmark functions are applied on HPSO-SA. In Section 5, HPSO-SA is used to solve the energy consumption problem in memory. In addition, simulation results are provided and compared with those of [11]. Conclusions and further research aspects are given in Section 6.

2. Background

2.1. Simulated Annealing Algorithm. SA [12] is a probabilistic variant of the local search method, which can, in contrast to PSO, escape local optima. SA is based on an analogy taken from thermodynamics which is as follows: In order to grow a crystal, we start by heating material until it reaches its molten state. Then, we reduce the temperature of this crystal melt gradually, until the crystal structure is formed. A standard SA procedure begins by generating an initial solution at random. At initial stages, a small random change is made in the current solution s_c . Then the objective function value of the new solution s_n is calculated and compared with that of the current solution. A move is made to the new solution if it has better value or if the probability function implemented in SA has a higher value than a randomly generated number. Otherwise a new solution is generated and evaluated. The probability of accepting a new solution is given as follows:

$$p = \begin{cases} 1, & \text{if } f(s_n) - f(s_c) < 0, \\ \exp\left(\frac{-|f(s_n) - f(s_c)|}{T}\right), & \text{otherwise.} \end{cases} \quad (1)$$

The calculation of this probability relies on a parameter T , which is referred to as temperature, since it plays a similar role as the temperature in the physical annealing process. To avoid getting trapped at a local minimum point, the rate of reduction should be slow. In our problem we use the following method to reduce the temperature: $\gamma = 0.99$ and

$$T_{i+1} = \gamma T_i, \quad (2)$$

where $i = 0, 1, \dots$

Thus, at the start of SA most worsening moves may be accepted, but in the end only improving ones are likely to be allowed, which can help the procedure jump out of a local minimum. The algorithm may be terminated after a certain volume fraction of the structure has been reached or after a prespecified runtime.

2.2. Particle Swarm Optimization. PSO is a population based stochastic optimization technique developed by [13], inspired by social behavior patterns of organisms that live and interact within large groups. In particular, it incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior.

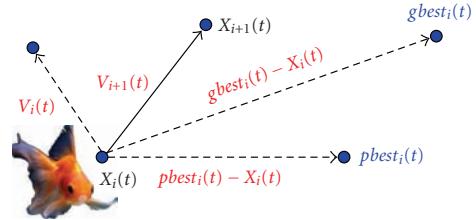


FIGURE 1: Movement of each particle.

PSO algorithm is based on an idea that particles move through the search space with velocities that are dynamically adjusted according to their historical behaviors. Therefore, the particles have the tendency to move towards the better and better search area over the course of search process. PSO algorithm starts with a group of random (or not) particles (solutions) and then searches for optima by updating each generation. Each particle is treated as a volume-less particle (a point) in the n -dimensional search space. The i th particle is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$. At each generation, the particles are updated by using following two *best* values.

- (i) The first value is the best solution (fitness) a particle has achieved so far (the fitness value is also stored). This value is called *pbest*.
- (ii) The second value is the best value tracked by the particle swarm optimizer so far (by any particle) in the population. This best value is a global best and is called *gbest*. When a particle takes part of a population as its topological neighbors, the best value is a local best and is called *lbest*.

At each iteration, these two best values are combined to adjust the velocity along each dimension, which is then used to compute a new iteration step for the particle. A portion of adjustment to the velocity is influenced by the individual's previous best position (*pbest*), considered as the *cognition* component, and this portion is influenced by the best in the neighborhood (*lbest* or *gbest*), the social component (see Figure 1). With the addition of the inertia factor, ω , by [14] (for balancing the global and the local search), the equations for velocity adjustment are

$$\begin{aligned} v_{i+1} = & \omega v_i + c_1 * \text{random}(0, 1) * (pbest_i - x_i), \\ & + c_2 * \text{random}(0, 1) * (gbest_i - x_i), \end{aligned} \quad (3)$$

$$x_{i+1} = x_i + v_{i+1}, \quad (4)$$

where $\text{random}(0, 1)$ is a random number independently generated within the range of $[0, 1]$ and c_1 and c_2 are two learning factors which control the influence of the social and cognitive components (usually, $c_1 = c_2 = 2$, see [15]).

In (3), if the sum on the right side exceeds a constant value, then the velocity on that dimension is assigned to be $V_i \min$ or $V_i \max$. Thus, particle velocities are clamped to the range of $[V_i \min; V_i \max]$, which serves as a constraint to control the global exploration ability of PSO algorithm.

This also reduces the likelihood of particles leaving the search space. Note that the values of x_i are not restricted to the range $[V_i \min; V_i \max]$; it only limits the maximum distance that a particle will move during one iteration.

3. HPSO-SA Hybrid Algorithm

This section presents a new hybrid HPSO-SA algorithm which combines the advantages of both PSO (that has a strong global-search ability) and SA (that has a strong local-search ability). Other applications of hybrid PSO and SA algorithm can be found [9, 10, 16–19].

This hybrid approach makes full use of the exploration capability of both PSO and SA and offsets the weaknesses of each. Consequently, through application of SA to PSO, the proposed algorithm is capable of escaping from a local optimum. However, if SA is applied to PSO at each iteration, the computational cost will increase sharply and at the same time the fast convergence ability of PSO may be weakened. In order to flexibly integrate PSO with SA, SA is applied to PSO every K iterations if no improvement of the global best solution does occur. Therefore, the hybrid HPSO-SA approach is able to keep fast convergence (most of the time) thanks to PSO, and to escape from a local optimum with the aid of SA. In order to allow PSO jump out of a local optimum, SA is applied to the best solution in the swarm found so far, each K iterations that is predefined to be 270 ~ 500 (based on our experimentations).

The hybrid HPSO-SA algorithm works as illustrated in Algorithm 1, where one has the following.

- (i) *Description of a Particle.* Each particle (solution) $X \in S$ is represented by its $n > 0$ components, that is, $X = (x_1, x_2, \dots, x_n)$, where $i = 1, 2, \dots, n$ and n represents the dimension of the optimization problem to solve.
- (ii) *Initial Swarm.* Initial Swarm corresponds to population of particles that will evolve. Each particle x_i is initialized with uniform random value between the lower and upper boundaries of the interval defining the optimization problem.
- (iii) *Evaluate Function.* Evaluate (or fitness) function in HPSO-SA algorithm is typically the objective function that we want to minimize in the problem. It serves for each solution to be tested for suitability to the environment under consideration.
- (iv) *SA Algorithm.* If no improvement of the global best solution occur during the last K iterations, then it means that the algorithm is trapped in a local optimum point. To escape out from local optimum, we apply SA algorithm to global best solution. The performance of SA depends on the definition of the several control parameters.
 - (a) *Initial Temperature.* Kirkpatrick [20] suggested that a suitable initial temperature is one that results in an average probability χ_0 of a solution that increases f being accepted of about 0.8. The value of T_0 will clearly depend on the scaling of f and, hence, be problem-specific. It can be estimated by conducting an initial search (100 iterations in next simulations) in which all increases in f are accepted and calculating the average objective increase observed δf . T_0 is then given by

$$T_0 = -\frac{\delta f}{\ln(\chi_0)}. \quad (5)$$

- (b) *Accept Function.* Function $\text{Accept}(\text{current_solution}, \text{Neighbor}, T)$ is decided by the acceptance probability given by (1), which is the probability of accepting configuration Neighbor .
- (c) *Generate Function.* The neighborhood of each solution x is generated by using the following Equation:

$$x \leftarrow x + d\sigma r, \quad (6)$$

where d is the direction of the new neighborhood and takes either 1 or -1 , σ is random number with Gaussian (0,1) distribution and r is a constant that correspond to the radius of neighborhood generator.

- (d) *SA_Stop_Criterion.* The stopping criterion of SA algorithm defines when the system has reached 3000 function evaluations or maximum number of functions evaluations or Optimal_solution are not attained.
- (e) *Decrementing the Temperature.* The most commonly used temperature reducing function is geometric (see (2)). In next simulations $\gamma = 0.99$.
- (f) *Inner-Loop.* The length of each temperature level determines the number $q = 150$ of solutions generated at each temperature, T .

4. Experiments Results

4.1. Benchmark Functions. In order to compare the performance of HPSO-SA hybrid algorithm with those described in [2, 8–10], we use benchmark functions [7] described in Table 1. These functions provide a good starting point for testing the credibility of an optimization algorithm. For each of these functions, there are many local optima in their solution spaces. The number of local optima increases with increasing complexity of the functions, that is, with increasing dimension. In the following experiments, we used 10-, 20- and 30-dimensional functions except in the case of Himmelblau and Shubert functions that are two-dimensional by definition (see Figure 2 for 3D representation).

4.2. Simulation Results and Discussions. To verify the efficiency and effectiveness of HPSO-SA hybrid algorithm, the experimental results of HPSO-SA approach are compared with those obtained by [2, 8–10]. Our HPSO-SA hybrid

```

(1) iter ← 0, cpt ← 0, Initialize swarm_size particles
(2) stop_criterion ← maximum number of function evaluations or Optimal_solution is not attained
(3) while Not stop_criterion do
(4)   for each particle  $i = 1$  to swarm_size do
(5)     Evaluate(particle( $i$ )) if the fitness value is better than the best fitness value (cbest) in history then
(6)       Update current value as the new cbest.
(7)   end
(8) end
(9) Choose the particle with the best fitness value in the neighborhood (gbest)
(10)  for each particle  $i = 1$  to swarm_size do
(11)    Update particle velocity according to Equation (3)
(12)    Enforce velocity bounds
(13)    Update particle position according to Equation (4)
(14)    Enforce particle bounds
(15)  end
(16)  if there is no improvement of global best solution then
(17)    cpt ← cpt + 1
(18)  end
(19)  Update global best solution
(20)  cpt ← 0
(21)  if cpt = K then
(22)    cpt ← 0
(23)    //Apply SA to global best solution
(24)    iterSA ← 0, Initialize T according to Equation (5)
(25)    current_solution ← global_best_solution
(26)    current_cost ← Evaluate(current_solution)
(27)    while Not SA_stop_criterion do
(28)      while inner-loop stop criterion do
(29)        Neighbor ← Generate(current_solution)
(30)        Neighbor_cost ← Evaluate(Neighbor)
(31)        if Accept(current_cost, Neighbor_cost, T) then
(32)          current_solution ← Neighbor
(33)          current_cost ← Neighbor_cost
(34)        end
(35)        iterSA ← iterSA + 1
(36)        Update (global_best_solution)
(37)      end
(38)      Update(T) according to Equation(2)
(39)      Update (SA_stop_criterion)
(40)    end
(41)  end
(42)  iter ← iter + 1, Update (stop_criterion)
(43) end

```

ALGORITHM 1: HPSO-SA hybrid algorithm.

algorithm is written in C++ and was compiled using gcc version 2.95.2 (Dev-cpp) on a laptop with Windows Vista x64 Premium Home Edition running Intel Core 2 Quad (Q9000) at 2 GHz and having 4 Gb memory.

4.2.1. Comparison with Results Obtained by Using TL-PSO Algorithm [8]. In this section we compare HPSO-SA approach with TL-PSO method [8] that is based on combining the excellence of both PSO and Tabu Search. As described in [8], we apply HPSO-SA algorithm to the four following benchmark problems: Rastrigin, Schwefel, Griewank and Rosenbrock function. Here, the number of particles in the swarm is 30. The number of dimension of

searching $n = 20$ and the number of objective function evaluations is 60000 (i.e., 2000×30). The results obtained after numerical simulations are shown in Table 2. These results indicate the Mean, Best, and Worst values obtained under the same condition over 50 trials. By analyzing Table 2, we conclude that the results obtained by HPSO-SA algorithm are preferable in comparison with those obtained by TL-PSO algorithm.

4.2.2. Comparison with Other PSO Algorithms Described in [2]. Performance of four Particle Swarm Optimization algorithms, namely classical PSO, Attraction-Repulsion based PSO (ATREPSO), Quadratic Interpolation based PSO

TABLE 1: Standard benchmark functions adopted in this work.

Function	Problem	Range	$f(x^*)$	Classification
Sphere	$\sum_{i=1}^n x_i^2$	$[-100; 100]$	0	Unimodal
Rastrigin	$\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12; 5.12]$	0	Multimodal
Griewank	$(1/4000) \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	$[-600; 600]$	0	Multimodal
Rosenbrock	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-2.048; 2.048]$	0	Unimodal
Quartic	$(\sum_{i=1}^n i x_i^4) + \text{rand}[0, 1]$	$[-1.28; 1.28]$	0	Noisy
Schwefel	$420.9687n - \sum_{i=1}^n (x_i \sin(\sqrt{ x_i }))$	$[-500.0; 500.0]$	0	Multimodal
Ackley	$20 + e - 20e^{-0.2((1/n) \sum_{i=1}^n x_i^2)^{1/2}} - e^{(1/n) \sum_{i=1}^n \cos(2\pi x_i)}$	$[-30.0; 30.0]$	0	Multimodal
Michalewicz	$-\sum_{i=1}^n \sin(x_i) \sin^{2m}((i - x_i^2)/\pi)$	$[-\pi; \pi]$	—	Multimodal
Himmelblau	$(x_2 + x_1^2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + x_1$	$[-5.0; 5.0]$	-3.78396	Multimodal
Shubert	$\sum_{i=1}^5 i \cos((i + 1)x_1 + i) \sum_{i=1}^5 i \cos((i + 1)x_2 + i)$	$[-10.0; 100]$	-186.7309	Multimodal

TABLE 2: Comparison of hybrid HPSO-SA algorithm with TL-PSO approach [8].

<i>Function</i>	Mean		Best		Worst	
	TL-PSO	HPSO-SA	TL-PSO	HPSO-SA	TL-PSO	HPSO-SA
<i>Rastrigin</i>	8.7161	5.23034e – 05	4.0589	4.35994e – 09	17.053	8.15835e – 04
<i>Schwefel</i>	445.8830	206.693255	0.6748	0	829.4431	355.318
<i>Griewank</i>	0.0228	1.18294e – 03	1.0507e – 5	1.2299e – 15	0.0739	0.0172263
<i>Rosenbrock</i>	19.5580	0.58359742	0.1331	0.0442625	71.5439	1.36171

TABLE 3: Comparison of mean/standard deviation of solutions obtained by using hybrid HPSO-SA algorithm and approaches described in [2].

<i>Function</i>	PSO	QIPSO	ATREPSO	GMPSO	HPSO-SA
Rastrigin	22.339158 15.932042	11.946888 9.161526	19.425979 14.349046	20.079185 13.700202	0 0
Sphere	1.167749e – 45 5.222331e – 46	0.000000 0.000000	4,000289e – 17 0.000246	7,263579e – 17 6.188854e – 17	5.3656e – 32 2.98492e – 31
Griewank	0.031646 0.025322	0.011580 0.012850	0.025158 0.028140	0.024462 0.039304	3,32255e – 20 2.68415e – 20
Rosenbrock	22.191725 1.615544e + 04	8.939011 3.106359	19.490820 3.964335e + 04	14.159547 4.335439e + 04	0,227048188 0.243978057
Noisy	8.681602 9.001534	0.451109 0.328623	8.046617 8.862385	7.160675 7.665802	0,002019998 0.000650347
Schwefel	-6178.559896 4.893329e + 02	-6355.586640 477.532584	-6183.677600 469.611104	-6047.670898 482.926738	-8379.66 2.20425e – 19
Ackley	3.483903e – 18 8.359535e – 19	2.461811e – 24 0.014425	0.018493 0.014747	1.474933e – 18 1.153709e – 08	7.43546e – 16 1.09382e – 15
Michalewicz	-18.159400 1.051050	-18.469600 0.092966	-18.982900 0.272579	-18.399800 0.403722	-19,62555806 0.00926944
Himmelblau	-3.331488 1.243290	-3.783961 0.190394	-3.751458 0.174460	-3.460233 0.457820	-3,78396 3.16001e – 15
Shubert	-186.730941 1.424154e – 05	-186.730942 0.000000	-186.730941 1.424154e – 05	-186.730942 1.525879e – 05	-186,730942 8.66746e – 14

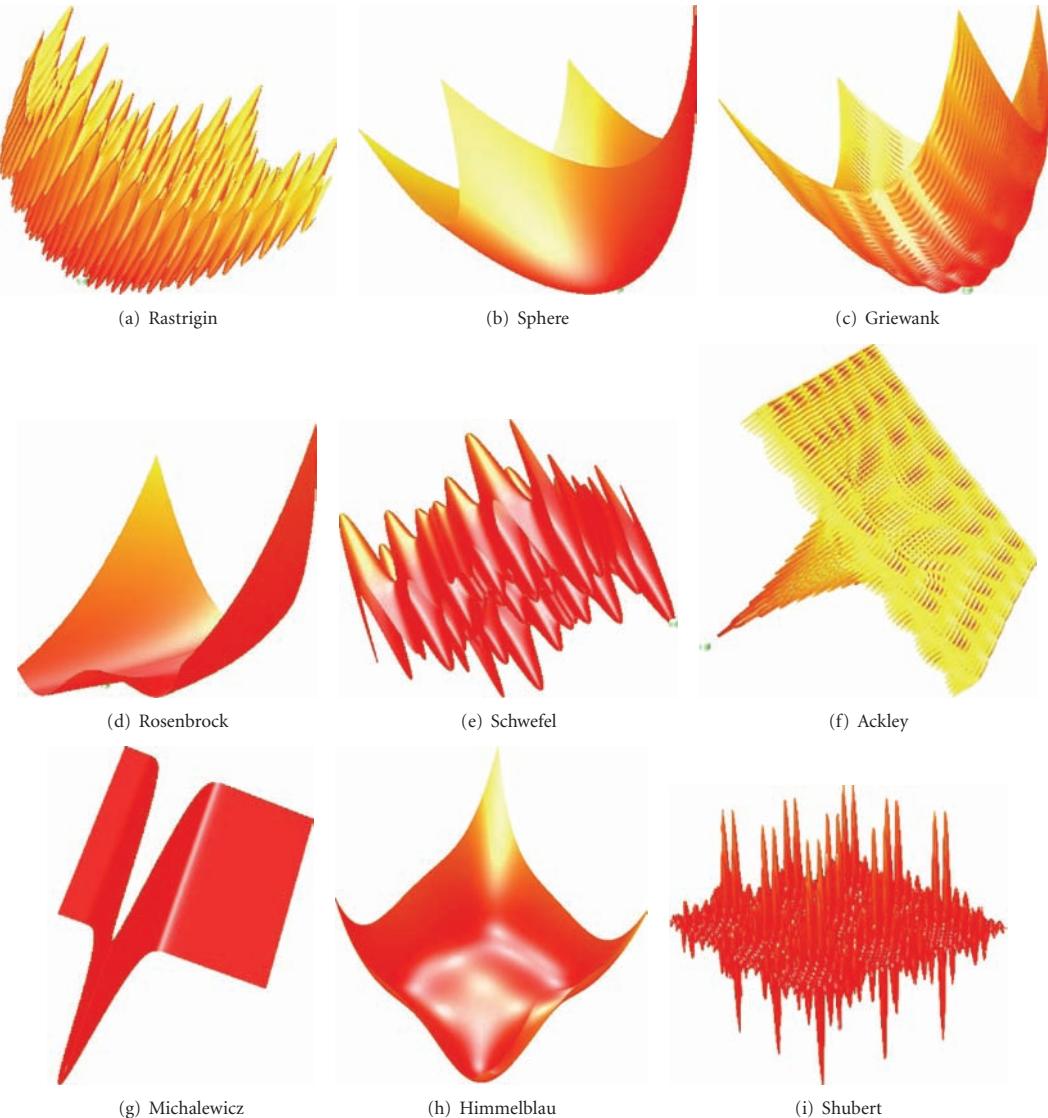


FIGURE 2: 3D mathematical benchmark functions.

(QIPSO) and Gaussian Mutation based PSO (GMPSO) is evaluated in [2]. The algorithms presented in this paper are guided by the diversity of the population to search the global optimal solution of a given optimization problem, where as GMPSO uses the concept of mutation and QIPSO uses the reproduction operator to generate a new member of the swarm.

In order to make a fair comparison between classical PSO, ATREPSO, QIPSO, GMPSO and HPSO-SA approach, we fixed, as indicated in [2], the same seed for random number generation so that the initial swarm population is same for all five algorithms. The number of particles in the swarm is 30. The algorithms use a linearly decreasing inertia

weight ω which starts at 0.9 and ends at 0.4, with the user defined parameters $c_1 = c_2 = 2.0$. For each algorithm, the number of objective function evaluations is 300000. A total of 30 runs for each experimental setting were conducted and the average fitness of the best solutions throughout the run was recorded. The mean solution and the standard deviation (note that the standard deviation indicates the stability of the algorithms), found by the five algorithms, is listed in Table 3. The numerical results given in Table 3 show the following.

- (i) All the algorithms outperform the classical Particle Swarm Optimization.

- (ii) HPSO-SA algorithm gives much better performances in comparison to PSO, QIPSO, ATREPSO, and GMPSO, out of the Sphere's and Ackley's functions.
- (iii) On Sphere's function, QIPSO obtains better results than those obtained by HPSO-SA approach. But when the maximum number of iterations is fixed to 1.5×10^6 , HPSO-SA obtains the optimal value.
- (iv) The analysis of the results, obtained for Ackley's function, shows that QIPSO obtains better mean result than HPSO-SA algorithm. However, HPSO-SA has a much smaller standard deviation.

4.2.3. Comparison with Other PSO Algorithms Described in [10]. In this section four benchmark functions are used to compare the relative performance of HPSO-SA algorithm with SUPER-SAPSO, SAPSO, and PSO algorithms described in [10].

For all comparisons, the number of particles was set to 30. HPSO-SA algorithm uses a linearly decreasing inertia weight ω which starts at 0.9 and ends at 0.4, with the user defined parameters $c_1 = c_2 = 2.0$, 20 runs are conducted for each experimental setting and, for each algorithm, the average value is given in Table 4.

In all above experiments, HPSO-SA algorithm obtains better results in comparison to those obtained by both the standard PSO and SAPSO algorithm [10]. A comparison of HPSO-SA algorithm and SUPER-SAPSO [10], shows that the last one converges faster than HPSO-SA.

SUPER-SAPSO uses an expression for the particle movements ($x_{t+1} = (x_t + v_{t+1})T$ where $T \ll 1$) which is well-adapted to the case where the global optimum is 0. This is the reason why SUPER-SAPSO needs a very small number of iterations in this case.

4.2.4. Comparison with PSO-SA Algorithm Described in [9]. In this section performances of HPSO-SA are compared with these of PSOSA [9], Genetic Algorithm and hybrid algorithm [21].

Table 5 lists different results obtained for three different dimensions of each function. The optimum value of Sphere, Rastrigin and Griewank was set to be $1e - 10$ and the goal value of Rosenbrock function was set to be $1e - 06$ (as indicated in [9]).

To make a fair comparison, the maximum number of function evaluations allowed was set to 20000, 30000 and 40000 for HPSO-SA and PSOSA algorithms when the number of particle was set to 20. HPSO-SA algorithm uses a linearly decreasing inertia weight ω which starts at 0.9 and ends at 0.4, with the user defined parameters $c_1 = c_2 = 2.0$.

The numerical results given in Table 5 show that:

- (i) Over four benchmark functions, HPSO-SA and PSOSA do better than standard GA and hybrid algorithm [21].
- (ii) For Sphere, Rastrigin and Griewank functions, HPSO-SA and PSOSA algorithms obtain optimal solutions within specified constraints (number of objective function evaluations).

- (iii) For Rosenbrock function, PSOSA obtains better results than HPSO-SA for dimension 20, but for dimensions 10 and 30, HPSO-SA does better and has smaller standard deviation.

5. Reducing Memory Energy Consumption in Embedded Systems

5.1. Description of the Memory Problem. According to trends in [22], memory will become the major energy consumer in an embedded system. Indeed, embedded systems must integrate multiple complex functionalities which needs bigger battery and memory. Hence, reducing memory energy consumption of these systems has never been as topical. In this paper, we will focus on software techniques for the memory management. In order to reduce memory energy consumption, most authors rely on Scratch-Pad Memories (SPMs) rather than caches [23]. Although cache memory helps a lot with program speed, it is not the appropriate for most of the embedded systems. In fact, cache increases the system size and its energy cost (cache area plus managing logic). Like cache, SPM consists of small, fast SRAM. The main difference is that SPM is directly and explicitly managed at the software level, either by the developer or by the compiler which makes it more predictable. SPM requires up to 40% less energy and 34% less area than cache [24]. In this paper, we will therefore use an SPM in our memory architecture. Due to the reduced SPM size, we allocate space for interesting data only whereas, the remaining is placed in main memory (DRAM). In order to determine interesting data, we use *data profiling* to gather memory access frequency information. The Tabu Search (TS) approach consists of allocating space for data in SPM based on TS principles [25]. More details about how TS is implemented can be found in [11].

In order to compute energy cost of the system, we propose an energy consumption estimation model, for our memory architecture composed by an SPM, an instruction cache and a DRAM. Equation (7) gives the energy model where the three terms refer to the total energy consumed, respectively, in SPM, in instruction cache and in DRAM.

$$E = E_{spmr} + E_{tic} + E_{tdram}. \quad (7)$$

In this model, we distinguish between the two cache write policies: Write-Through (WT) and Write-Back (WB). In a WT cache, every write to the cache causes a synchronous write to DRAM. Alternatively, in a WB cache, writes are not immediately mirrored to DRAM. Instead, the cache tracks which of its locations have been written over and then, it marks these locations as dirty. The data in these locations is written back to DRAM when that data is evicted from the cache [26]. In this paper, the aim is to minimize the energy for the detailed estimation model presented as follows:

$$E = N_{spmr} * E_{spmr} \quad (8)$$

$$+ N_{spmw} * E_{spmw} \quad (9)$$

TABLE 4: Performance results of HPSO-SA, SUPER-SAPSO, SAPSO, and PSO algorithms on benchmark functions.

Function	Algorithm	Number of iterations	Average Error
Rastrigin	PSO	2989	0.097814
	SAPSO	2168	0.07877
	SUPER-SAPSO	5	0.0
	HPSO-SA	2022.43	0.0
Sphere	PSO	805	0.094367
	SAPSO	503	0.085026
	SUPER-SAPSO	4	0.0
	HPSO-SA	501.15	7, 92e - 32
Griewank	PSO	2004	0.082172
	SAPSO	1517	0.075321
	SUPER-SAPSO	3	0.0
	HPSO-SA	1496.35	3, 61e - 20
Ackley	PSO	4909	0.099742
	SAPSO	3041	0.099461
	SUPER-SAPSO	5	0.0
	HPSO-SA	3020.65	8, 92e - 16

TABLE 5: Performance comparison between PSOSA, Sid.GA, Hybrid and HPSO-SA for benchmark functions [9].

Function	Dimension	PSOSA	Std.GA	Hybrid	HPSO-SA
Sphere	10	0 ± 0	2.43e - 04 ± 1.14e - 05	2.42e - 04 ± 2.17e - 05	0 ± 0
	20	0 ± 0	0.00145 ± 6.22e - 05	0.00212 ± 2.75e - 04	0 ± 0
	30	0 ± 0	0.00442 ± 1.78e - 04	0.01203 ± 6.33e - 04	0 ± 0
Rastrigin	10	0 ± 0	3.1667 ± 0.2237	3.0599 ± 0.1535	0 ± 0
	20	0 ± 0	16.8732 ± 0.6007	11.6590 ± 0.3602	0 ± 0
	20	0 ± 0	49.3212 ± 1.1204	27.8119 ± 0.8059	0 ± 0
Griewank	10	0 ± 0	283.251 ± 1.812	0.09078 ± 0.03306	0 ± 0
	20	0 ± 0	611.266 ± 3.572	0.00459 ± 0.01209	0 ± 0
	30	0 ± 0	889.537 ± 3.939	0.09911 ± 0.00106	0 ± 0
Rosenbrock	10	0.17856 ± 1.25988	109.810 ± 6.212	43.521 ± 16.047	0.146 ± 0.224193
	20	0.00043 ± 0.00111	146.912 ± 10.951	169.112 ± 21.535	0.246897 ± 0.231982
	30	0.57431 ± 4.05976	199.730 ± 16.285	187.033 ± 22.960	0.439149 ± 0.304347

$$\begin{aligned}
& + \sum_{k=1}^{N_{icr}} [h_{ik} * E_{icr} + (1 - h_{ik}) \\
& * [E_{dramr} + E_{icw} + (1 - WP_i) \\
& * DB_{ik} * (E_{icr} + E_{dramw})]]
\end{aligned} \tag{10}$$

$$\begin{aligned}
& + \sum_{k=1}^{N_{icw}} [WP_i * E_{dramw} + h_{ik} * E_{icw} + (1 - WP_i) \\
& * (1 - h_{ik}) * [E_{icw} + DB_{ik} * (E_{icr} + E_{dramw})]]
\end{aligned} \tag{11}$$

$$+ N_{dramr} * E_{dramr} \tag{12}$$

$$+ N_{dramw} * E_{dramw}. \tag{13}$$

Equations (8) and (9) represent, respectively, the total energy consumed during reading and writing from/to SPM. Equations (10) and (11) represent, respectively, the total energy consumed during reading and writing from/to instruction cache. When, Equations (12) and (13) represent, respectively, the total energy consumed during reading and writing from/to DRAM. The various terms used in this energy model are explained in Table 6.

As SPM has got a lot of advantages, it is clearly preferable to put as much data as possible in it. In other words, we must maximize terms N_{spmr} and N_{spmw} in the model. Hence, the problem becomes to maximize the number of accesses to the SPM. It is therefore a combinatorial optimization problem like knapsack problem [27]. We want to fill SPM that can hold a maximum capacity of C with some combination of data from a list of N possible data each with $size_i$ and $access\ number_i$ so that the access number of the data allocated into SPM is maximized. This problem has a single linear constraint, a linear objective function which sums

TABLE 6: List of terms.

Term	Meaning
E_{spmr}	Energy consumed during a reading from SPM.
E_{spmw}	Energy consumed during a writing into SPM.
N_{spmr}	Reading access number to SPM.
N_{spmw}	Writing access number to SPM.
E_{icr}	Energy consumed during a reading from instruction cache.
E_{icw}	Energy consumed during a writing into instruction cache.
N_{icr}	Reading access number to instruction cache.
N_{icw}	Writing access number to instruction cache.
E_{dramr}	Energy consumed during a reading from DRAM.
E_{dramw}	Energy consumed during a writing into DRAM.
N_{dramr}	Reading access number to DRAM.
N_{dramw}	Writing access number to DRAM.
WP_i	The considered cache write policy: WT or WB. In case of WT, $WP_i = 1$ else in case of WB then $WP_i = 0$.
DB_i	Dirty Bit used in case of WB to indicate during the access k if the instruction cache line has been modified before ($DB_i = 1$) or not ($DB_i = 0$).
h_{ik}	Type of the access k to the instruction cache. In case of cache hit, $h_{ik} = 1$. In case of cache miss, $h_{ik} = 0$.

the sizes of the data allocated into SPM, and the added restriction that each data will be in the SPM or not. If N is the total number of data, then a solution is just a finite sequence s of N terms such that $s[n]$ is either 0 or the size of the n th data. $s[n] = 0$ if and only if the n th data is not selected in the solution. This solution must satisfy the constraint of not exceeding the maximum SPM capacity (i.e., $\sum_{i=1}^N s[i] \leq C$).

5.2. Discrete Sequential Hybrid HPSO-SA Algorithm. This section should be considered as an attempt to use hybrid evolutionary algorithms for reducing energy consumption in embedded systems. Here, the focus is on the use of HPSO-SA algorithm designed in previous sections. Since the problem under consideration is discrete and has specific features, HPSO-SA needs changes.

solution (Particle). A solution can be represented by an array having size equal to the number of the data. Each element from this array denotes whether a data is included in the SPM ("1") or not ("0"). The HPSO-SA algorithm starts with an initial swarm which is randomly initialized.

Evaluate Function. It is the objective function that we want to minimize in the problem. It serves for each solution to be tested for suitability to the environment under consideration

$$\begin{aligned} \text{Evaluate}(solution) = & \text{Total_Number_Access_all_data} \\ & - \text{Number_Access}(solution). \end{aligned} \quad (14)$$

Position Update Equation. Each dimension j of the particle i is updated by using (15):

$$x_{ij} = \begin{cases} 1, & \text{rand} < \text{sigm}(v_{ij}), \\ 0, & \text{otherwise}, \end{cases} \quad (15)$$

where $\text{sigm}(v_{ij})$ is the sigmoid function, used to scale the velocities between 0 and 1, defined as:

$$\text{sigm}(v_{ij}) = \frac{1}{1 + \exp(-v_{ij})} \quad (16)$$

Generate Function. SA uses a notion of neighborhood relation. Let S be the set of all feasible solutions to the problem and $f : S \rightarrow \mathcal{R}$ the objective function to be minimized. A neighborhood relation is a binary relation $N \subseteq S \times S$ with some desired properties. The interpretation of $N(s, s')$ is that solution s is a neighbor of solution s' in the search space of all solutions S . A neighbor heuristic proceeds in steps. Starting search at some initial solution s_0 and then each step moves from the current solution to some neighbor according to rules specific to the heuristic. At each iteration, SA algorithm generates a random neighbor of the *current_solution* (line 10). The neighborhood relation is defined as follows:

- (1) with probability equal to 0.03, the value of each element of *current_solution* is flipped from 1 to 0 or from 0 to 1;
- (2) validate solution:
while *Not feasible* (*current_solution* must satisfy the constraint of not exceeding the maximum SPM capacity.) (*current_solution*) do
Remove the data j having a low number of access from *current_solution*: (*current_solution*[j] $\leftarrow 0$).

Accept Function. the key idea in the SA approach is the function *Accept* which specifies the probability of accepting the move from *current_solution* to a *Neighbor* solution, which also depends on so called temperature (T). The function *Accept* should satisfy the following conditions:

- (1) $p = 1$ if solution *Neighbor* is better than *current_solution* in terms of the cost function f : (i.e., $f(\text{Neighbor}) < f(\text{current_solution})$ in a minimization problem);
- (2) if *Neighbor* is worse than *current_solution* the value of p is positive (i.e., it allows for moving to a worse solution), but decreases with $|f(\text{Neighbor}) - f(\text{current_solution})|$;
- (3) for fixed *current_solution* and *Neighbor*, when *Neighbor* is worse than *current_solution* the value of p decreases with time and tends to 0.

The function $\text{Accept}(C_{cost}, N_{cost}, T)$ is decided by the probability of accepting configuration *Neighbor*. This probability is given by the following formula:

$$p = \begin{cases} 1, & N_{cost} < C_{cost}, \\ \frac{1}{2} \text{rand} * (1 + e^{(C_{cost} - N_{cost})/T}), & \text{otherwise}, \end{cases} \quad (17)$$

where T is the temperature and rand is a random number independently generated within the range of $[0, 1]$.

5.3. Discrete Cooperative Distributed Hybrid HPSO-SA Algorithm. For Distributed hybrid HPSO-SA (*HPSO-SA_Dist*) algorithm, we use independent subswarms of particles with their own fitness functions which evolve in isolation, except for an exchange of some particles (migration). A set of $m = 30$ particles is assigned to each of the P processors, for a total population size of $m \times P$. The set assigned to each processor is its subswarm. The processors are connected by an interconnection network with a ring topology. Initial subswarms consist of a randomly constructed assignment created at each processor. Each processor, disjointly and in parallel, executes the *HPSO-SA_Seq* algorithm on its subswarm for a certain number of generations. Afterwards, each subswarm exchanges (*HPSO – SA_Dist* runs in *Asynchronous* mode: at the 100th iteration, each processor sends its best solution and continues the improvement of its subswarm and verifies if it does not receive a solution from its neighbor.) its best particle (migrant) with its neighbors. We exchange the particles themselves (i.e., the migrant is removed from one subswarm and added to another). Hence, the size of the subswarm remains the same after migration (the worst particle is removed). The process continues with the separate improvement of each current solution for a maximum number of iterations. At the end of the process the best solution that exists constitutes the final assignment.

5.4. Experimental Results. In order to compute the energy cost of studied memory architecture composed by an SPM, an instruction cache and a DRAM, we proposed an energy consumption estimation model which is explained in [11]. Hybrid HPSO-SA algorithms and TS have been implemented on a cluster of PCs running under Windows XP Professional version 2002. The cluster is composed by 4 Pentium (D) machines running at 3 GHz. Each processor has 1 Gbyte of memory. Table 7 gives a description of the benchmarks used and they also can be downloaded from [28].

In experiments, 30 different executions for each heuristic are performed and the best and average results obtained on these 30 executions are recorded. In this case, the best and the average solutions give similar results. Figure 3 shows that both *HPSO-SA_Seq* and *HPSO-SA_Dist* achieve better performances than TS on energy savings. In fact, hybrid HPSO-SA heuristics consume from 76.23% (StatemateCE) to 98.92% (ShaCE) less energy than TS.

As *HPSO-SA_Seq* and *HPSO-SA_Dist* give similar results, we decide to experiment their behavior when considering their execution time. We recorded the average execution times needed by *HPSO-SA_Seq* and *HPSO-SA_Dist* (running on a cluster of 4 PCs) to achieve the 30 executions. Figure 4 presents the results obtained on the largest (size) benchmarks. From this figure, we see that the Distributed HPSO-SA version (*HPSO-SA_Dist*) is faster than the Sequential HPSO-SA version (*HPSO-SA_Seq*). In fact, *HPSO-SA_Dist*

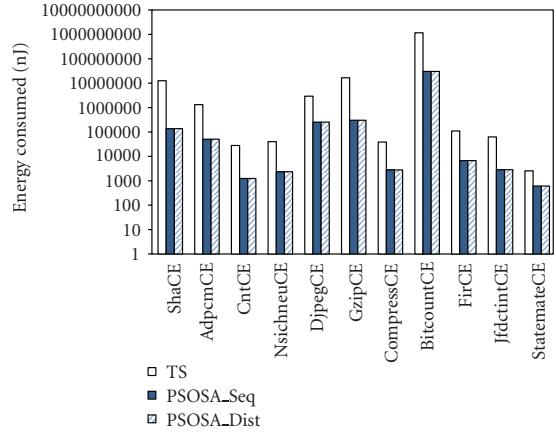


FIGURE 3: Energy consumed by benchmarks studied in this work.

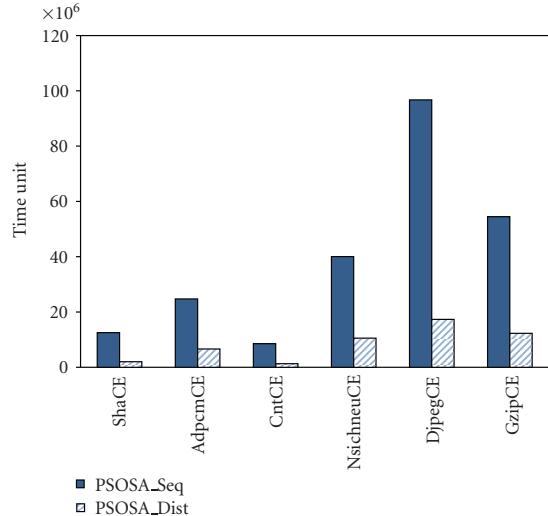


FIGURE 4: Execution time used by HPSO-SA algorithms on benchmarks studied in this work.

requires 73.16% (AdpcmCE) to 84.65% (CntCE) less execution time than *HPSO-SA_Seq*. The Distributed HPSO-SA version is always faster than the Sequential HPSO-SA version.

6. Conclusion and Perspectives

In this paper, we have designed a hybrid algorithm (HPSO-SA) that combines the exploration ability of PSO with the exploitation ability of SA, and is capable of preventing premature convergence. Compared with QIPSO, ATREPSO and GMPSO [2], TL-PSO [8], PSO-SA [9] and SUPER-SAPSO [10] on well-known benchmark functions and for the problem of reducing energy consumption in embedded systems memories, it has been shown that HPSO-SA performs well in terms of accuracy, convergence rate, stability and robustness. In future, we will also compare the performances

TABLE 7: List of Benchmarks.

Benchmark	Suite	Description
ShaCE	MiBench	The secure hash algorithm that produces a 160-bit message digest for a given input.
BitcountCE	MiBench	Tests the bit manipulation abilities of a processor by counting the number of bits in an array of integers.
FirCE	SNU-RT	Finite impulse response filter (signal processing over a 700 items long sample).
JfdctintCE	SNU-RT	Discrete-cosine transformation on 8×8 pixel block.
AdpcmCE	Mälardalen	Adaptive pulse code modulation algorithm.
CntCE	Mälardalen	Counts nonnegative numbers in a matrix.
CompressCE	Mälardalen	Data compression using lzw.
DjpegCE	Mediabenchs	JPEG decoding.
GzipCE	Spec 2000	Compression.
NsicneuCE	Wcet Benchs	Simulate an extended Petri net.
StatemateCE	Wcet Benchs	Automatically generated code with more than 250 if-statements.
		Automatically generated code.

of HPSO-SA with the above mentioned algorithms on the embedded systems memory saving problem.

In addition, we will compare HPSO-SA algorithm with other hybrid algorithms (PSO-GA, PSO-MDP, PSO-TS) whose design is in progress by the authors. Comparison will also be done on additional benchmark functions and more complex problems including functions with dimensionality larger than 30.

Acknowledgments

The authors are grateful to anonymous referees for their pertinent comments and suggestions. Dawood Khan helped the authors with the intricacies of the English language. The work of M. Idrissi Aouad is supported by the French national research agency (ANR) in the Future Architectures program.

References

- [1] M. Locatelli, "Simulated annealing algorithms for continuous global optimization," in *Handbook of Global Optimization*, P. M. Pardalos and H. E. Romeijn, Eds., vol. 2, pp. 179–230, Kluwer Academic, 2001.
- [2] M. Pant, R. Thangaraj, and A. Abraham, "Particle swarm based metaheuristics for function optimization and engineering applications," in *Proceedings of the 7th Computer Information Systems and Industrial Management Applications (CISIM '08)*, vol. 7, pp. 84–90, IEEE Computer Society, Washington, DC, USA, 2008.
- [3] J. Kennedy and C. E. Russell, "Swarm intelligence," in *Morgan Kaufmann*, Academic Press, 2001.
- [4] A. M. Marco, T. Stützle et al., "Convergence behavior of the fully informed particle swarm optimization algorithm," in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08)*, pp. 71–78, 2008.
- [5] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, 2005.
- [6] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization. An overview," in *Swarm Intelligence*, vol. 1, pp. 33–57, 2007.
- [7] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," Tech. Rep. 2005005, Nanyang Technological University, Singapore; IIT Kanpur, India, 2005.
- [8] S. Nakano, A. Ishigame, and K. Yasuda, "Consideration of particle swarm optimization combined with tabu search," *IEEJ Transactions on Electronics, Information and Systems*, vol. 128, pp. 1162–1167, 2008, Special Issue on 'The Electronics, Information and Systems Conference Electronics, Information and Systems Society, I.E.E. of Japan'.
- [9] G. Yang, D. Chen, and G. Zhou, "A new hybrid algorithm of particle swarm optimization," in *Lecture Notes in Computer Science*, vol. 4115, pp. 50–60, 2006.
- [10] M. Bahrepour, E. Mahdipour, R. Cheloi, and M. Yaghoobi, "Super-sapso: a new sa-based pso algorithm," in *Applications of Soft Computing*, vol. 58, pp. 423–430, 2009.
- [11] M. I. Aouad, R. Schott, and O. Zendra, "A tabu search heuristic for scratch-pad memory management," in *Proceedings of the International Conference on Software Engineering and Technology (ICSET '10)*, vol. 64, pp. 386–390, WASET, Rome, Italy, 2010.
- [12] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [13] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE Computer Society, 1995.
- [14] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '98)*, pp. 69–73, IEEE Computer Society, 1998.
- [15] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, pp. 1945–1950, 1999.
- [16] W. J. Xia and Z. M. Wu, "A hybrid particle swarm optimization approach for the job-shop scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 29, no. 3–4, pp. 360–366, 2006.
- [17] D. Chaojun and Z. Qiu, "Particle swarm optimization algorithm based on the idea of simulated annealing," *International Journal of Computer Science and Network Security*, vol. 6, no. 10, pp. 152–157, 2006.

- [18] L. Fang, P. Chen, and S. Liu, "Particle swarm optimization with simulated annealing for tsp," in *Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED '07)*, pp. 206–210, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wis, USA, 2007.
- [19] X. Wang and J. Li, "Hybrid particle swarm optimization with simulated annealing," in *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics (ICMLC '04)*, vol. 4, pp. 2402–2405, 2004.
- [20] S. Kirkpatrick, "Optimization by simulated annealing: quantitative studies," *Journal of Statistical Physics*, vol. 34, no. 5-6, pp. 975–986, 1984.
- [21] L. Morten, K. R. Thomas, and T. Krink, *Hybrid Particle Swarm Optimization with Breeding and Subpopulations*, Springer, Berlin, Germany, 2000.
- [22] "ITRS, System Drivers," 2007, http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_SystemDrivers.pdf.
- [23] M. I. Aouad and O. Zendra, "A survey of scratch-pad memory management techniques for low-power and -energy," in *Proceedings of the 2nd ECOOP Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOLPS '07)*, pp. 31–38, Berlin, Germany, 2007.
- [24] H. B. Fradj, A. El Ouadihi, C. Belleudy, and M. Auguin, "Energy aware memory architecture configuration," *SIGARCH Computer Architecture News*, vol. 33, no. 3, pp. 3–9, 2005.
- [25] M. Gendreau, *An Introduction to Tabu Search*, vol. 57, Kluwer Academic, Boston, Mass, USA, 2003.
- [26] A. Tanenbaum, *Architecture de l'ordinateur*, 5th edition, 2005.
- [27] U. P. H. Kellerer and D. Pisinger, *Knapsack Problems*, Springer, Berlin, Germany, 2004.
- [28] "Benchmarks," <http://www.loria.fr/~idrissma/benchs.zip>.

SEQUENTIAL AND DISTRIBUTED HYBRID GA-SA ALGORITHMS FOR ENERGY OPTIMIZATION IN EMBEDDED SYSTEMS

Lhassane IDOUMGHAR^{1,2}, Maha IDRISI AOUAD¹, René SCHOTT³ and Olivier ZENDRA¹

¹ INRIA Nancy - Grand Est / LORIA. 615, rue du Jardin Botanique, 54600 Villers-Lès-Nancy, France

Email: {Maha.IdrissiAouad, Olivier.Zendra}@inria.fr

² LMIA - MAGE, Université de Haute-Alsace. 4, rue des Frères Lumière, 68093 Mulhouse, France

Email: lhassane.idoumghar@uha.fr

³ IECN - LORIA, Nancy-Université, Université Henri Poincaré. 54506 Vandoeuvre-Lès-Nancy, France

Email: Rene.Schott@loria.fr

ABSTRACT

Reducing memory energy consumption in embedded systems is crucial. In this paper, we propose new hybrid sequential and distributed algorithms based on Simulated Annealing (SA) and Genetic Algorithms (GA) in order to reduce memory energy consumption in embedded systems. Our algorithms outperform the Tabu Search (TS) approach. In fact, our hybrid algorithms manage to consume nearly from 76% up to 98% less memory energy than TS. Execution time savings for the distributed version (nearly from 72% up to 74% for a cluster of 4 PCs) are also recorded.

KEYWORDS

Distributed systems, embedded systems, Genetic Algorithms, memory management, optimization, Simulated Annealing.

1. INTRODUCTION

Embedded systems have become more energy greedy due to technology evolution. Indeed, they must integrate multiple complex functionalities which needs bigger battery and memory. Accordingly, memory will become the major energy consumer in an embedded system (ITRS, 2007). Hence, reducing this memory energy is crucial. In this paper, we will focus on software optimizations working on the memory management.

In order to save energy, most authors rely on Scratch-Pad Memories (*SPMs*) rather than cache memories and various related research directions have been investigated (Idrissi Aouad and Zendra, 2007). Although caches, help a lot with program speed, they are not the most appropriate for embedded systems. In fact, caches increase the system size and its energy cost because of cache area plus managing logic. Like cache, SPM consists of small, fast SRAM, but the main difference is that SPM is directly and explicitly managed at the software level, either by the developer or by the compiler, whereas cache requires extra dedicated circuits. Its software management makes it more predictable as we avoid cache miss cases. Compared to cache, SPM thus has several advantages (Idrissi Aouad and Zendra, 2007). SPM requires up to 40% less energy and 34% less area than cache (Ben Fradj et al., 2005). Additionally, manufacturing SPM cost is lower and a large variety of chips with SPM is available in the market (Adiletta et al., 2002). We will therefore use an SPM in our memory architecture.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 gives the memory energy model we used. Section 4 describes our optimization problem. Section 5 presents our hybrid GA-SA algorithms. Section 6 shows the experimental results. Section 7 concludes and gives some perspectives.

2. RELATED WORKS AND EXISTING HEURISTICS

Due to the reduced SPM size, authors try to optimally allocate data in it in order to realize energy savings. An approach is to place interesting data (the most frequently accessed (Wehmeyer et al., 2004), the most cache-conflict prone (Panda et al., 1997), etc.) in SPM (a low access cost memory) whereas the other data will be placed in main memory (*DRAM*) (a low storage cost memory). In order to determine interesting data, authors use *data profiling* to gather memory access frequency information. This information can be collected either statistically by analyzing the source code or dynamically by profiling the application (data size, access frequency, etc.). Thus, most effective strategies are:

Allocate data into SPM by access number/size (BEH): data are sorted according to their ratio (access number/size). The data with the highest ratio is allocated first into SPM as there is space available else it is allocated in DRAM. This heuristic uses a sorting method which can be computationally expensive for a large amount of data. Additionally, BEH will not work very well in a dynamic perspective where the SPM maximum capacity is not known in advance.

Allocate data into SPM based on Tabu Search (TS): details about how TS is implemented can be found in (Idrissi Aouad et al., 2010). TS is as energy efficient as BEH as shown in (Idrissi Aouad et al., 2010). In contrast, TS is easy to implement and since no sorting is necessary, unlike BEH, the corresponding time is saved. In addition to that, in a dynamic perspective where the SPM maximum capacity is not known in advance, TS will perform better than BEH.

3. MEMORY ENERGY ESTIMATION MODEL

In order to compute the energy cost of the system, we propose in this section an energy consumption estimation model for our considered memory architecture composed by an SPM, an instruction cache and a DRAM. Equation 1 gives the energy model where the three terms refer to the total energy consumed respectively in SPM, in instruction cache and in DRAM.

$$E = E_{tspm} + E_{tic} + E_{tdram} \quad (1)$$

In this model, we distinguish between the two cache write policies: Write-Through (WT) and Write-Back (WB). In a WT cache, every write to the cache causes a synchronous write to DRAM. Alternatively, in a WB cache, writes are not immediately mirrored to DRAM. Instead, the cache tracks which of its locations have been written over and then, it marks these locations as dirty. The data in these locations is written back to DRAM when those data are evicted from the cache (Tanenbaum, 2005). Our aim is to minimize the detailed energy estimation model presented below:

$$E = N_{spmr} * E_{spmr} \quad (2)$$

$$+ N_{spmw} * E_{spmw} \quad (3)$$

$$+ \sum_{k=1}^{N_{icr}} [h_{i_k} * E_{icr} + (1 - h_{i_k}) * [E_{dramr} + E_{icw} + (1 - WP_i) * DB_{i_k} * (E_{icr} + E_{dramw})]] \quad (4)$$

$$+ \sum_{k=1}^{N_{icw}} [WP_i * E_{dramw} + h_{i_k} * E_{icw} + (1 - WP_i) * (1 - h_{i_k}) * [E_{icw} + DB_{i_k} * (E_{icr} + E_{dramw})]] \quad (5)$$

$$+ N_{dramr} * E_{dramr} \quad (6)$$

$$+ N_{dramw} * E_{dramw} \quad (7)$$

Lines (2) and (3) represent respectively the total energy consumed during a reading and during a writing from/into SPM. Lines (4) and (5) represent respectively the total energy consumed during a reading and during a writing from/into instruction cache. When, lines (6) and (7) represent respectively the total energy consumed during a reading and during a writing from/into DRAM. The various terms used in this energy model are explained in Table 1.

Table 1. List of terms.

Term	Meaning
E_{spmr}	Energy consumed during a reading from SPM.
E_{spmw}	Energy consumed during a writing into SPM.
N_{spmr}	Reading access number to SPM.
N_{spmw}	Writing access number to SPM.
E_{icr}	Energy consumed during a reading from instruction cache.
E_{icw}	Energy consumed during a writing into instruction cache.
N_{icr}	Reading access number to instruction cache.
N_{icw}	Writing access number to instruction cache.
E_{dramr}	Energy consumed during a reading from DRAM.
E_{dramw}	Energy consumed during a writing into DRAM.
N_{dramr}	Reading access number to DRAM.
N_{dramw}	Writing access number to DRAM.
WP_i	The considered cache write policy: WT or WB. In case of WT, $WP_i = 1$ else in case of WB then $WP_i = 0$.
DB_{i_k}	Dirty Bit used in case of WB to indicate during the access k if the instruction cache line has been modified before ($DB_i = 1$) or not ($DB_i = 0$).
h_{i_k}	Type of the access k to the instruction cache. In case of cache hit, $h_{i_k} = 1$. In case of cache miss, $h_{i_k} = 0$.

4. OPTIMIZATION PROBLEM

Our problem is a combinatorial optimization problem. It is a kind of knapsack problem (Kellerer et al., 2004). We want to fill SPM that can hold a maximum capacity of C with some combination of data from a list of N possible data each with $size_i$ and $access\ number_i$ so that the access number of the data allocated into SPM is maximized. This problem has a single linear constraint, a linear objective function which sums the sizes of the data allocated into SPM, and the added restriction that each data will be in the SPM or not. If N is the total number of data, then a solution is just a finite sequence s of N terms such that $s[n]$ is either 0 or the size of the n_{th} data. $s[n] = 0$ if and only if the n_{th} data is not selected in the solution. This solution must satisfy the constraint of not exceeding the maximum SPM capacity (*i.e.* $\sum_{i=1}^N s[i] \leq C$).

5. OUR HYBRID GA-SA ALGORITHMS

For preliminaries on SA principles see (Kirkpatrick et al., 1983) and for background on GAs principles see (Sivanandam and Deepa, 2007).

5.1 Our Sequential Hybrid GA-SA Algorithm

The principles of our Sequential hybrid GA-SA (*GASA_Seq*) algorithm are described in Algorithm 1.

```

1 Initialize  $p_m, p_c, p_i \in [0,1]$  and  $i \leftarrow 1$ 
2  $maxGen \leftarrow 10000$ 
3 Generate population  $P_0$ 
4 Evaluate  $P_0$  and find the best solution  $\pi^*$ 
5  $\pi_{Elite} \leftarrow \pi^*$ 
6  $stop\_criterion \leftarrow \text{false}$ 
7 while Not  $stop\_criterion$  do
8    $P_i \leftarrow \emptyset$ 
9   for  $j := 1$  to  $PopSize/2$  do
10    | Select two parents  $p_1$  and  $p_2$  from  $P_{i-1}$ 
11    | offspring  $\leftarrow (p_1, p_2)$ 
12    | With probability  $p_c$ , perform offspring := crossover( $p_1, p_2$ )
13    | With probability  $p_m$ , mutate offspring
14    | With probability  $p_i$ , improve offspring by using SA (with
       |  $maxiter = 200$ )
15    | Evaluate offspring and add it to  $P_i$ 
16   end
17   Add  $P_{i-1}$  to  $P_i$ 
18   Sort  $P_i$ 
19   Keep the  $PopSize$  best solution in  $P_i$ 
20   Find the best solution  $\pi^*$  in  $P_i$ 
21   if  $\pi^*$  is better than  $\pi_{Elite}$  then
22     |  $\pi_{Elite} \leftarrow \pi^*$ 
23   end
24   if  $fitness(\pi_{Elite}) = 0$  OR  $i = maxGen$  then
25     |  $stop\_criterion \leftarrow \text{true}$ 
26   end
27   Update(i)
28 end

```

Algorithm 1. Sequential hybrid GA-SA algorithm.

5.2 Our Distributed Hybrid GA-SA Algorithm

The principles of our Distributed hybrid GA-SA (*GASA_Dist*) algorithm are described in Algorithm 2.

```

1 Initialize  $p_m, p_c, p_i \in [0,1]$  and  $i \leftarrow 1$ 
2  $maxGen \leftarrow 2500$ 
3 Generate population  $P_0$ 
4 Evaluate  $P_0$  and find the best solution  $\pi^*$ 
5  $\pi_{Elite} \leftarrow \pi^*$ 
6  $stop\_criterion \leftarrow \text{false}$ 
7 while Not  $stop\_criterion$  do
8    $P_i \leftarrow \emptyset$ 
9   for  $j := 1$  to  $PopSize/2$  do
10    | Select two parents  $p_1$  and  $p_2$  from  $P_{i-1}$ 
11    | offspring  $\leftarrow (p_1, p_2)$ 
12    | With probability  $p_c$ , perform offspring := crossover( $p_1, p_2$ )
13    | With probability  $p_m$ , mutate offspring
14    | With probability  $p_i$ , improve offspring by using SA (with
       |  $maxiter = 200$ )
15    | Evaluate offspring and add it to  $P_i$ 
16   end
17   Add  $P_{i-1}$  to  $P_i$ 
18   if Migrate_condition then
19     | Receive  $n = 5$  Individuals and add its to  $P_i$ 
20   end
21   Sort  $P_i$ 
22   Keep the  $PopSize$  best solution in  $P_i$ 
23   Find the best solution  $\pi^*$  in  $P_i$ 
24   if  $\pi^*$  is better than  $\pi_{Elite}$  then
25     |  $\pi_{Elite} \leftarrow \pi^*$ 
26   end
27   if  $fitness(\pi_{Elite}) = 0$  OR  $i = maxGen$  then
28     |  $stop\_criterion \leftarrow \text{true}$ 
29   end
30   Update(i)
31 end

```

Algorithm 2. Distributed hybrid GA-SA algorithm running on each processor.

We use independent subpopulations of individuals with their own fitness functions which evolve in isolation, except for an exchange of some individuals (migration). A set of $m = 30$ individuals is assigned to each of the P processors, for a total population size of $m * P$. The set assigned to each processor is its subpopulation. The processors are connected by an interconnection network with a ring topology. Initial subpopulations consist of a randomly constructed assignment created at each processor. Each processor, separately and in parallel, executes the *GASA_Seq* algorithm on its subpopulation for a certain number of generations. Afterwards, each subpopulation exchanges a specific number of individuals (migrants) with its neighbors. We exchange the individuals themselves, i.e. the migrants are removed from one subpopulation and added to another. Hence the size of the population remains the same after migration. The process continues with the separate evolution of each subpopulation for a certain number of generations. At the end of the process the best individual that exists constitutes the final assignment.

$$Fitness(solution) = Total_Number_Access_all_data - Number_Access(solution) \quad (8)$$

- **Initial subpopulation:** initial population P_0 is created randomly. In Algorithms 1 and 2, $PopSize = 30$ is the size of every population P_i . During each of the $maxGen$ generations, $PopSize$ offsprings are generated through the crossover of parents selected from the subpopulation.
- **Fitness evaluation:** the fitness function (see Equation 8) in hybrid GA-SA algorithms is typically the objective function we want to minimize in the problem. It serves for each individual to be tested for suitability to the environment under consideration.
- **Selection operator:** in our approach, we use a random selection that is a simple method for implementing fitness-proportionate selection. On each stage, an individual is randomly selected to be in the pool of parents for the next generation.
- **Crossover operator:** the crossover is a random process defined by a probability $p_c = 0.6$ and applied sequentially to pairs of parents chosen randomly in the population. It consists in exchanging parts of the genetic material of the parents in order to create two childes (offspring). In our approach we have used the two points crossover where two points are chosen randomly and the contents between these points are exchanged between two mated parents.
- **Mutation operator:** mutation operator changes the new offspring by flipping bits from 1 to 0 or from 0 to 1 (while solution remains feasible). Mutation operator can occur at each bit position in the array with some probability (in our algorithm $p_m = 0.06$).
- **Improve operator:** with probability $p_i = 0.01$, SA algorithm is applied to the new offspring.

6. EXPERIMENTAL RESULTS

For our experiments, we consider a memory architecture composed by an SPM, an instruction cache and a DRAM. Our energy model is based on the OTAWA framework (Cassé and Rochange, 2007) to collect information about number of accesses and on the energy consumption estimation tool CACTI (Wilton and Jouppi, 1996) in order to collect information about energy per access to each memory kind. Our presented GA-SA algorithms and TS have been implemented with the C++ language on a PC Pentium (D), with a 3 GHz processor and 1 Gbyte of memory running under Windows XP Professional version 2002. For the GA-SA distributed version, we used a cluster of 4 PCs having the same characteristics and the MPICH2 (version 1.0.7) library for communication across the processes. The machines were running their normal daily loads in addition to our algorithms. Table 2 presents the benchmarks used. They also can be downloaded from (Benchmarks, 2010). Due to the lack of space, just the WB cache policy results are plotted. Same amounts are recorded for the WT mode ($E_{WTmode} \neq E_{WBmode}$).

Table 2. List of Benchmarks.

Benchmarks	Suite	Description
ShaCE	MiBench	The secure hash algorithm that produces a 160-bit message digest for a given input.
BitcountCE	MiBench	Tests the bit manipulation abilities of a processor by counting

		the number of bits in an array of integers.
FirCE	SNU-RT	Finite impulse response filter (signal processing algorithms) over a 700 items long sample.
JfdctintCE	SNU-RT	Discrete-cosine transformation on 8x8 pixel block.
AdpcmCE	Mälardalen	Adaptive pulse code modulation algorithm.
CntCE	Mälardalen	Counts non-negative numbers in a matrix.
CompressCE	Mälardalen	Data compression using lzw.
DjpegCE	Mediabenchs	JPEG decoding.
GzipCE	Spec 2000	Compression.
NsichneuCE	Wcet Benchs	Simulate an extended Petri net. Automatically generated code with more than 250 if-statements.
StatemateCE	Wcet Benchs	Automatically generated code.

We investigate the energy performances of our hybrid GA-SA algorithms when compared to TS. 30 different executions for each heuristic were performed and the best and average results were recorded. Both *GASA_Seq* and *GASA_Dist* always find the best solution and give similar results. We therefore call *GASA* the common value of their executions. *GASA_v1* refers to the hybrid GA-SA algorithm that uses SA as the third reproduction operator. In contrast, *GASA_v2* starts by applying GA (without SA operator in Algorithm 1, $p_i = 0$) until the algorithm completely finishes. Then, ten individuals from the final population are randomly selected and SA algorithm is executed over them. Figure 1 presents the results obtained. Both GA-SA heuristics achieve better performances than TS. In fact, they consume from 76.23% (StatemateCE) up to 98.92% (ShaCE) less energy than TS.

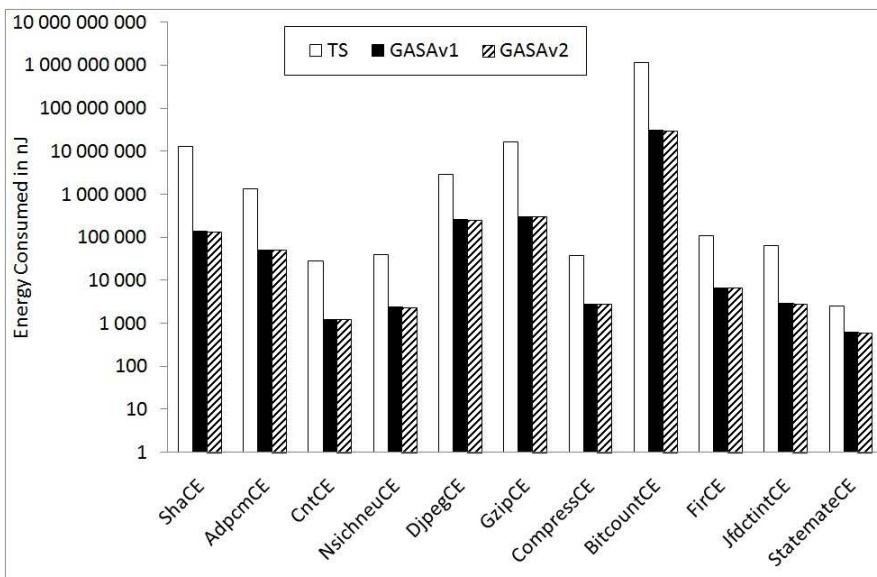


Figure 1. Energy consumed by our benchmarks with WB mode.

We also record the average execution times needed by *GASA_Seq* and *GASA_Dist* to achieve the 30 executions. Figure 2 presents the results obtained on the largest (size) benchmarks. From this figure, we can notice that both distributed GA-SA versions (*GASA_v1_Dist* and *GASA_v2_Dist*) are faster than the sequential GA-SA versions (*GASA_v1_Seq* and *GASA_v2_Seq*). In fact, *GASA_v1_Dist* requires from 72.31% (GzipCE) up to 74.67% (CntCE) less execution time than *GASA_v1_Seq* on one hand. On the other hand, *GASA_v2_Dist* needs from 72.72% (DjpegCE) up to 74.64% (CntCE) less execution time than *GASA_v2_Seq*.

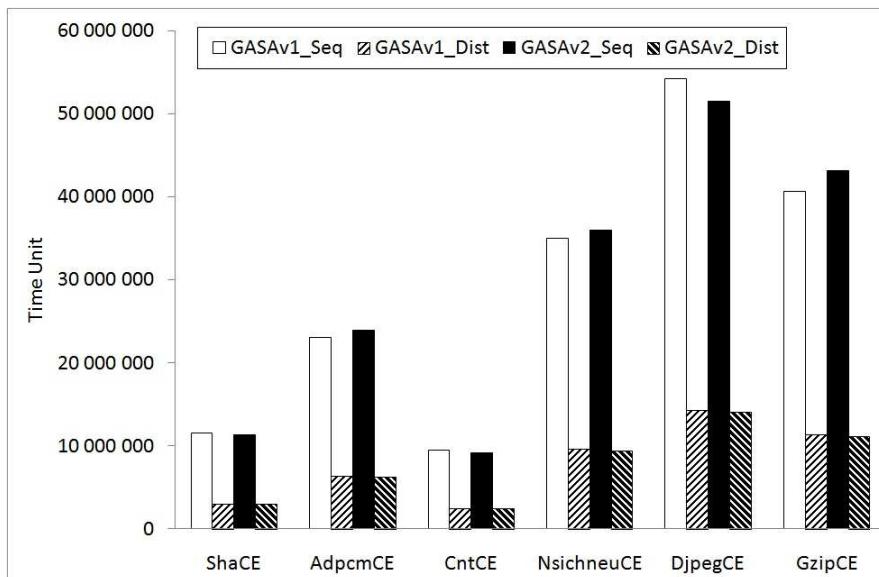


Figure 2. Execution time used by our GA-SA algorithms.

7. CONCLUSION AND PERSPECTIVES

We have proposed new hybrid sequential and distributed algorithms based on SA and GAs in order to reduce memory energy consumption in embedded systems. Our GA-SA algorithms consume nearly from 76% up to 98% less memory energy than TS. Execution time savings for the distributed GA-SA version (nearly from 72% up to 74% for a cluster of 4 PCs) are also recorded. In future work, we plan to investigate the same problem when relaxing the memory constraints by considering random memory sizes on one hand. On the other hand, we plan to explore evolutionary heuristics.

ACKNOWLEDGEMENT

The work of M. Idrissi Aouad and O. Zendra is supported by the french national research agency (ANR) in the Future Architectures program.

REFERENCES

- Adiletta, M. et al, 2002. The Next Generation of Intel IXP Network Processors. *In Intel Technology Journal*, Vol. 6, No. 3, pp. 6-18.
- Benchmarks, 2010. www.loria.fr/~idrissma/benchs.zip.
- Ben Fradj, H. et al, 2005. Energy Aware Memory Architecture Configuration. *In ACM SIGARCH Computer Architecture News*, Vol. 33, No. 3, pp. 3-9.
- Cassé, H. and Rochange, C., 2007. OTAWA, Open Tool for Adaptable WCET Analysis. *Proceedings of Design, Automation and Test in Europe (DATE)*. Nice, France. Poster session.
- Idrissi Aouad, M. et al, 2010. A Tabu Search Heuristic for Scratch-Pad Memory Management. *Proceedings of International Conference on Software Engineering and Technology (ICSET)*. Rome, Italy, Vol. 64, pp. 386-390. WASET Publisher.
- Idrissi Aouad, M. and Zendra, O., 2007. A Survey of Scratch-Pad Memory Management Techniques for low-power and -energy. *Proceedings of 2nd ECOOP Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOLPS)*. Berlin, Germany, pp. 31-38.
- ITRS, 2007. System Drivers. http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_SystemDrivers.pdf.

- Kellerer, H. et al, 2004. *Knapsack Problems*. Springer Publisher, Berlin, Germany.
- Kirkpatrick et al, 1983. Optimization by Simulated Annealing. In *Science*, Vol. 220, pp. 671-680.
- Panda, P. R. et al, 1997. Efficient Utilization of Scratch-Pad Memory in Embedded Processor Applications. *Proceedings of European Design and Test Conference (EDTC)*. Washington, DC, USA, pp. 7.
- Sivanandam, S. N. and Deepa, S. N., 2007. *Introduction to Genetic Algorithms*. Springer Publishing Company, Incorporated.
- Tanenbaum, A., 2005. *Architecture de l'ordinateur 5^e édition*. Pearson Education.
- Wehmeyer, L. et al, 2004. Compiler-Optimized Usage of Partitioned Memories. *Proceedings of the 3rd Workshop on Memory Performance Issues (WMPI)*. Munich, Germany, pp. 114-120.
- Wilton, S. and Jouppi, N., 1996. Cacti: An Enhanced Cache Access and Cycle Time Model. In *IEEE Journal of Solid-State Circuits*, Vol. 31, pp. 677-688.

Simulated Annealing Algorithm for the Optimization of an Electrical Machine

L. Idoumghar¹, D. Fodorean², A.N'diaye³, D. Bouquain³, A. Miraoui³

¹ LMIA-MAGE and LORIA Université de Haute Alsace (Mulhouse, France) - lhassane.idoumghar@oha.fr

² Technical University of Cluj-Napoca (Cluj, Romania) - daniel.fodorean@mae.utcluj.ro

³ University of Technology from Belfort-Montbéliard (Belfort, France) - abdoul-ousman.n-diaye@utbm.fr,

davide.bouquain@utbm.fr, abdellatif.miraoui@utbm.fr

Abstract—This paper presents a novel Simulated Annealing (SA) algorithm for the optimization of an electrical machine. Firstly, the proposed SA algorithm is validated on five standard benchmark mathematical functions. Furthermore, the SA algorithm will be employed for the optimization of a brushless synchronous motor. The optimized topology of the motor will be analyzed numerically and the results will be validated experimentally. The obtained performances show that the proposed SA algorithm outperforms other algorithms described in the scientific literature.

Index Terms—Optimization, Simulated Annealing, Brushless Synchronous Motor, Numerical and Experimental Validation.

I. INTRODUCTION

From trains, to home appliance, or from automobiles to any industrial machine, the electrical machines represent the core of electromechanical systems, since all the movements and displacements are (or are indenting to be in the nearest future) employed based on electrical machines. Thus, their performances have significant influence on the behaviour of the entire system. For example, if one considers the electric/hybrid vehicle, it is widely recognized that one of the most important aspects for an automobile is its autonomy. Off course, the major drawback is related to the energy storage capability, but the machine's energetic performances play a crucial role for the whole system autonomy. Even if one uses the best control unit equipment and the most efficient energy storage device, and *if the electrical machine has poor energetic parameters*, the global system will have bad performances. Also, no one needs to have an electrical machine with good energetic performances, and which is over-sized. The investment will increase significantly and the whole system performances will be affected, because over-weight automobile involves more electric power consumption, and so decreased autonomy. Even in industrial environment the trend is to classify the electrical machines in different categories, based on their rated efficiency. Thus, the researchers are trying to exploit the limits of all category of electrical machines, or to propose new topologies, with the best

energetic performances and costly efficient. In this way, the optimization approach plays a crucial role in the design of electrical machines.

Over the last decades, various metaheuristics algorithms have been developed in order to solve complex search and optimization problems. Among these algorithms, a special place is occupied by Simulated Annealing (SA) [1,2] and optimization algorithms that make use of social or evolutionary behaviours like Particle Swarm Optimization (PSO) [3,4], Ant Colony Optimization (ACO) [5,6] and Genetic Algorithm [7,8,9,10].

SA is a probabilistic variant of the local search method, but it can, in contrast, escape local optima [11]. SA is based on an analogy taken from thermodynamics: to grow a crystal, we start by heating a row of material to a molten state. Next, we reduce the temperature of this crystal melt gradually, until the crystal structure is frozen in.

A standard SA procedure begins by generating an initial solution at random. At initial stages, a small random change is made in the current solution s_c . Then the objective function value of the new solution s_n is calculated and compared with that of the current solution. A move is made to the new solution if it has better value or if the probability function implemented in SA has a higher value than a randomly generated number. Otherwise a new solution is generated and evaluated. The probability of accepting a new solution is given as follows:

$$p = \begin{cases} 1 & \text{if } f(s_n) < f(s_c) \\ \exp\left(-\frac{f(s_n)-f(s_c)}{T}\right) & \text{otherwise} \end{cases} \quad (1)$$

The calculation of this probability relies on a temperature parameter T , which is referred to as temperature, since it plays a similar role as the temperature in the physical annealing process. To avoid getting trapped at a local minimum point, the rate of reduction should be slow. In our problem the following method, to reduce the temperature, has been used:

$$T_{i+1} = \gamma T_i \quad (2)$$

where $i = 0, 1, \dots$ and $\gamma = 0.99$.

Thus, at the start of SA most worsening moves may be accepted, but at the end only improving ones are likely to be allowed. This can help the procedure jump out of a local minimum. The algorithm may be terminated if a certain volume fraction of the structure has been reached or after a pre-specified run time. Additional concepts about SA and some of its applications can easily be found in literature [1,2,12,13,14].

This paper is organized as follows. Section II describes the application, meaning the outer-rotor permanent magnet synchronous machine (OR-PMSM). Section III, is devoted to SA algorithm used in this work. Section IV presents simulations results of the employed SA algorithm on five benchmark functions (to prove its operability) and the results of the optimized OR-PMSM. The simulation results are compared with those obtained in the scientific literature [3,5,18]. The simulation results of the optimized machine are compared with experimental ones, in Section V. Conclusions and further research aspects are given in Section VI.

II. DESCRIPTION OF AN OUTER-ROTOR PERMANENT MAGNET MACHINE

The studied machine is a brushless synchronous machine excited through permanent magnets (PMs). To be more specific, the authors have studied an outer-rotor permanent magnet synchronous machine (OR-PMSM). It should be mentioned that the optimization approach can be adapted to any other topology of electrical machines.

The main data of the designed OR-PMSM are: 1.5 kW output power, 650 rpm rated speed, 48 V_{cc} supplying voltage (battery voltage), 22 Nm rated torque and 24.78 A rated current. The winding of the OR-PMSM is of fractioned type, with 51 slots, and the number of poles is 46. The cross-section of the studied motor, as well as the main geometrical parameters used in the optimization process, can be identified in Figure 1.

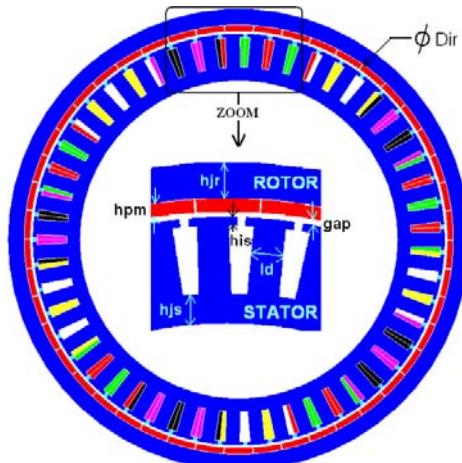


FIGURE 1. Cross-section of the studied OR-PMSM.

The goal of the paper is not to present the designing process of the machine, but to emphasize the main steps of the optimization process with some details related to the machine's geometry and performances, and some constraints used in the optimization process.

The motor design is based on magnetic reluctance equivalent circuit and the main parameters of the OR-PMSM. The goal of the optimization (and finally the objective function) is to obtain a lower mass for the machine, at the same output performances. From the designing process, we extract the main geometrical parameters which have an important influence on motor's performances, see Table I.

TABLE I: THE MAIN PARAMETERS USED FOR MACHINE'S OPTIMIZATION

Symbol	Description	Value	Variation limits
Dir	Rotor inner diameter	207 mm	[185; 210] mm
hjr	Rotor yoke height	8 mm	[5; 10] mm
his	Tooth isthmus height	2 mm	[1; 3] mm
hjs	Stator yoke height	6.5 mm	[5; 10] mm
ld	Tooth width	7 mm	[4; 10] mm
gap	Air-gap length	1 mm	[0.5; 1.5] mm
hmp	PM height	3 mm	[2.5; 7] mm
Lm	Machine's length	50 mm	[30; 80] mm
Supplementary performances constraints ("C")			
Tm	Motor torque	22 Nm	[21.8; 22.2] Nm
Pout	Output power	1500 W	[1490; 1510] W
Is	Current source	24.78 A	[15; 25] A

Also, the last column of Table I contain the variation limits of the main geometrical parameters, which will establish the limits of the space of solutions for the objective function. In Table I supplementary constraints were added, with regard to OR-PMSM performances expectations, meaning: the desired torque, output power and current absorption. The objective function to optimize is the mass of the active parts of the machine, here called m_{tot} . The total mass of the active parts of the machine needs to be minimized. m_{tot} represents the sum of all active parts of the machine:

$$m_{tot} = m_{copper} + m_{stat} + m_{rot} + m_{PM} \quad (3)$$

where m_{copper} is the mass of copper used for windings, m_{stat} is the mass of the stator core (teeth and yoke), m_{rot} is the mass of the rotor core and m_{PM} is the mass of the PMs.

The designed machine has the mass, energetic performances and total cost given in Table II (results obtained before optimization).

TABLE II: THE MAIN PERFORMANCES OF THE DESIGNED OR-PMSM

Symbol	Description	Value
m_{tot}	Total mass of the machine's active part	7.1207 kg
P_{fe}	Iron loss	34.2259 W
P_{co}	Copper loss	103.2254 W
η	Efficiency of the machine	0.9120
$\cos\phi$	Power factor of the machine	0.8739
c_{tot}	Total cost of the machine's active part	83.1590 €

The OR-PMSM will be optimized based on the proposed SA algorithm, having the following objective function:

$$\text{Minimize } f(\text{Dir}, \text{hjr}, \text{his}, \text{hjs}, \text{ld}, \text{gap}, \text{hmp}, \text{Lm}) = -P_{out} / m_{tot} + \text{penalty} \quad (4)$$

with $\text{penalty} = 0$ if all optimization constraints C given in Table I are respected. Otherwise,

$penalty = 10^6 \times \sum_{i=1}^3 |C_i - Limit_i|$, where:

$$Limit_i = \begin{cases} Lower\ Bound_i & \text{if } C_i < Lower\ Bound_i \\ Upper\ Bound_i & \text{if } C_i > Upper\ Bound_i \end{cases}$$

III. DESCRIPTION OF OUR SA ALGORITHM

The principle of our SA algorithm is described as follows:

```

iter ← 0
initialize T
stop_criterion ← maximum number of iterations
Initialize current_solution
Best_solution ← current_solution
Current_cost ← Evaluate(current_solution)
Best_cost ← current_cost
while Not stop_criterion do
    while inner-loop stop_criterion do
        Neighbour ← Generate(current_solution)
        Neighbour cost ← Evaluate(Neighbour)
        if Accept(current_cost, Neighbour_cost, T) then
            current_solution ← Neighbour
            current_cost ← Neighbour_cost
        end
        Update (best solution, iter)
    end
    Update(T) according Equation 2
    Update (stop_criterion)
end

```

where:

- Initial Temperature: a suitable initial temperature is one that results in an average probability p_0 of a solution that increases f being accepted of about 0.8. The value of T_0 will clearly depend on the scaling of f and, hence, be problem-specific. It can be estimated by conducting an initial search (100 iterations in next simulations) in which all increases in f are accepted and calculating the average objective increase observed δf . T_0 is then given by:

$$T_0 = -\frac{\delta f}{\ln p_0}$$

- Initial solution: Each solution $X \in S$ is represented by its $n > 0$ components, i.e., $X = (x_1, x_2, \dots, x_n)$, where $i = 1, 2, \dots, n$ and n represents the dimension of the optimization problem to solve. The SA algorithm starts with an initial solution which is initialized with uniform random value between the lower and upper boundaries of the interval defining the optimization problem.
- Variable best solution (*best_solution*) keeps the best solution found,

- Variable *iter* counts the iterations of the main loop.
- Evaluate function: Evaluate function (Eq. 4) in SA algorithm is typically the objective function that we want to minimize. It serves for each solution to be tested for suitability to the environment under consideration. In section 3, we will describe several objective functions that we have used in this work
- The main loop proceeds until the number of iterations exceeds maximum number of iterations.
- Generate function: Simulated annealing is a randomized algorithm for searching the space of feasible solutions using a notion of neighbourhood relation. Let S be the set of all feasible solutions to the problem and $f: S \rightarrow \mathbb{R}$ the objective function to be minimized. A neighbourhood relation is a binary relation $N \subseteq S \times S$ with some desired properties. The interpretation of $N(s,s')$ is that solution s is a neighbour of solution s' in the search space of all solutions S . A neighbour heuristic proceeds in steps. It starts searching at some initial solution s' and in each step moves from the current solution to some neighbour according to some rules specific to the heuristic. At each iteration our SA algorithm generates a random neighbour of the current solution.

In our case we define the neighbourhood relation as follows. Any two solutions $s, s' \in S$ are neighbours $N(s,s')$ if they differ by $m \in \{1,2,3\}$ terms randomly chosen between n .

- Accept function: The key idea in the SA approach is the function *Accept* which specifies the probability of accepting the move from current solution to a Neighbour solution, which also depends on so called temperature (T). The function *Accept* should satisfy the following conditions:
 - ✓ $p = 1$ if solution Neighbour is better than current solution (*current_solution*) in terms of the cost function f (i.e. $f(\text{Neighbour}) < f(\text{current_solution})$ in a minimization problem).
 - ✓ if Neighbour is worse than current solution the value of p is positive (i.e. it allows for moving to a worse solution), but decreases with $|f(\text{Neighbour}) - f(\text{current_solution})|$.
 - ✓ for fixed current solution and Neighbour, when Neighbour is worse than current solution the value of p decreases with time and tends to 0.

The function $\text{Accept}(c_{cost}, n_{cost}, T)$ is decided by the probability of accepting configuration Neighbour. This probability is given by the following formula:

$$p = \begin{cases} 1 & \text{if } n_{cost} < c_{cost} \\ \text{rand} \times \left(1 + \exp\left(\frac{c_{cost}-n_{cost}}{T}\right)\right) < 2.0 & \text{otherwise} \end{cases} \quad (5)$$

where:

- $c_{\text{cost}} = f(\text{current_solution})$
- $n_{\text{cost}} = f(\text{Neighbour})$
- T is the temperature
- rand is a random number independently generated within the range of [0,1].

IV. SIMULATION RESULTS

IV.1. BENCHMARK FUNCTIONS

Before validating all results obtained for OR-PMSM problem, we present in this section the performance of our SA algorithm with those described in [3,5,15,16]. In this work we use five benchmark functions [19] described in Table III. These functions provide a good launch pad for testing the credibility of an optimization algorithm. For these functions, there are many local optima in their solution spaces. The amount of local optima increases with increasing complexity of the functions, i.e. with increasing dimension. In our experiments, we used 20 and 30-dimensional functions.

TABLE III. FIVE BENCHMARK PROBLEMS ADOPTED IN THIS WORK.

Function	Problem	Range	Fitness of global optima
Rastrigin	$\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[-5.12, 5.12]	0
Rosenbrock	$\sum_{i=1}^{n-1} 100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	[-30, 30]	0
Schwefel	$-\sum_{i=1}^n (x_i \sin \sqrt{ x_i })$	[-500, 500]	-418.9829 n
Noisy	$\left(\sum_{i=1}^n (i+1) x_i^4 \right) + \text{rand}[0,1]$	[-1.28, 1.28]	0
Michalewicz	$\sum_{i=1}^n \sin(x_i) \sin^{2m} \left(\frac{i-x_i^2}{\pi} \right), m = 10$	[-\pi, \pi]	(not defined for n=20 or n=40)

IV.1.1. Comparison with results obtained by using Ant Colony paradigm [5,15,16]

In this section we will compare our SA algorithm with three of the existing continuous-domain ACO algorithms: Multivariate Ant Colony Algorithm for Continuous Optimization (MACACO) [5], Continuous Ant Colony System (CACS) [15] and Ant Colony Optimization in \mathbb{R}^n (ACO_R) [16].

As described in [5], we apply our algorithm to the three following benchmark problems: Rastrigin's, Rosenbrock's and Schwefel's function (dimension of searching $n = 30$) which are highly multimodal problems that represent a challenging task in continuous optimization.

For each benchmark problem considered in this section, we have, in Table IV, the mean and standard deviation produced by each algorithm after 30 independent runs.

TABLE IV. MEAN AND STANDARD DEVIATION VALUES OF THE BEST SOLUTION OBTAINED BY USING OUR SA ALGORITHM AND APPROACHES DESCRIBED IN [5] (DIMENSION $N = 30$ AND POPULATION OF 1000 ANTS WAS DEFINED FOR BOTH CACS AND MACACO ALGORITHMS).

Function	ACO_R	CACS	MACACO	SA
Rastrigin	101.65	57.17	0.00058	8.00 E-08
	± 21.01	± 14.14	± 0.00009	± 2.59369 E-08
Rosenbrock	29.93	1.23	7.48	0.384
	± 35.14	± 1.91	± 11.68	± 0.149790826
Schwefel	-8703.26	-8934.57	-12569.49	-12569.487
	± 721.53	± 633.78	± 0.01	± 0

Each algorithm should be allowed a maximum of 10^6 function evaluations for all the benchmark problems studied here, except for the Rosenbrock function, which could be evaluated up to $6 \cdot 10^6$ times. Note that in [5] a population of 1000 ants was defined for both CACS and MACACO algorithms.

By analysing Table IV we can see:

- that MACACO algorithm outperforms the CACS and ACO_R in two of the three functions, being worse than CACS on the Rosenbrock's function.
- that the results obtained by our SA algorithm are excellent in comparison with those obtained by ACO_R , MACACO and CACS algorithms. Our SA algorithm has a much smaller standard deviation for Rastrigin's and Rosenbrock's function and found the global optimum for Schwefel's function.

IV.1.2. Comparison with PSO algorithm described in [3]

Performance of four Particle Swarm Optimization (PSO) algorithms, namely classical PSO, Attraction-Repulsion based PSO (ATREPSO), Quadratic Interpolation based PSO (QIPSO) and Gaussian Mutation based PSO (GMPSO) are evaluated in [3].

In order to make a fair comparison of classical PSO, ATREPSO, QIPSO, GMPSO and our SA approach we fixed, as indicated in [3], for each algorithm, the maximum number of iterations allowed was set to 10000. The number of particles, used by the four algorithms, in the swarm is 30. A total of 30 runs for each experimental setting were

conducted and the average fitness of the best solutions throughout the run was recorded. The mean solution and the standard deviation¹ found by the five algorithms are listed in Table V.

The numerical results given in Table V show that:

- all the algorithms outperform the classical Particle Swarm Optimization.

TABLE V. COMPARISON OF MEAN/STANDARD DEVIATION OF SOLUTIONS OBTAINED BY USING OUR SA ALGORITHM AND PSO APPROACHES DESCRIBED IN [3] (DIMENSION N = 20 AND SWARM SIZE = 30).

Function	PSO	QIPSO	ATREPSO	GMPSO	SA
Rastrigin	22.339158	11.946888	19.425979	20.079185	1.924398 E-06
	± 15.932042	± 9.161526	± 14.349046	± 13.700202	± 8.40864 E-07
Rosenbrock	22.191725	8.939011	19.490820	14.159547	0.840377
	± 1.615544 E+04	± 3.106359	± 3.964335 E+04	± 4.335439 E+04	± 0.673092017
Schwefel	-6178.559896	-6355.586640	-6183.677600	-6047.670898	-8379.658
	± 4.893329 E+02	± 477.532584	± 469.611104	± 482.926738	± 1.86624 E-12
Noisy	8.681602	0.451109	8.046617	7.160675	9.363985 E-04
	± 9.001534	± 0.328623	± 8.862385	± 7.665802	± 4.77749 E-04
Michalewicz	-18.159400	-18.469600	-18.982900	-18.399800	-19.634785
	± 1.051050	± 0.092966	± 0.272579	± 0.403722	± 0.002062516

- SA algorithm gives much better performances in comparison to PSO, QIPSO, ATREPSO and GMPSO.
- the analysis of the results obtained by our approach for all five benchmark functions, studied in this work, shows the stability of our algorithm (i.e. SA algorithm has a much smaller standard deviation).

IV.2. OR-PMSM OPTIMIZATION

In this section we will present the optimization results that are need to be validated. Note that, in the field of electrical machines, there are two ways, generally accepted, for the validation of designed/optimized machines: by employing the numerical computation, based on finite element method (FEM), or by experimental validation. Both approaches will be verified in this paper. For the FEM analysis, the Flux2D software was used.

The SA optimization approach proposed by authors was employed in the design of the studied OR-PMSM. The algorithm which computes m_{tot} is based on the same analytical approach which gave us obtained the original solution, presented in Table II.

In Figure 2 are presented the variation of the main geometrical parameters, as well as the OR-PMSM main performances. The algorithm starts from a random solution which will be continuously improved until the maximum

¹ Note that the standard deviation indicates the stability of the algorithms

number of iterations was reached. Here, 3700 iterations were considered; beyond this limit, after several launch of the algorithm, no improvement has been found. For the sake of presentation, in Figure 2 only each 30th solution was plotted (if all 3700 solutions are shown, the plot becomes crowded and it will be very difficult to have an idea about the real results). It is noticed that the torque, output power and current constraints are respected. Also, the energetic performances are satisfactory. For a given geometrical configuration the best obtained mass is of 3.76 kg. This result should be verified numerically and finally through tests.

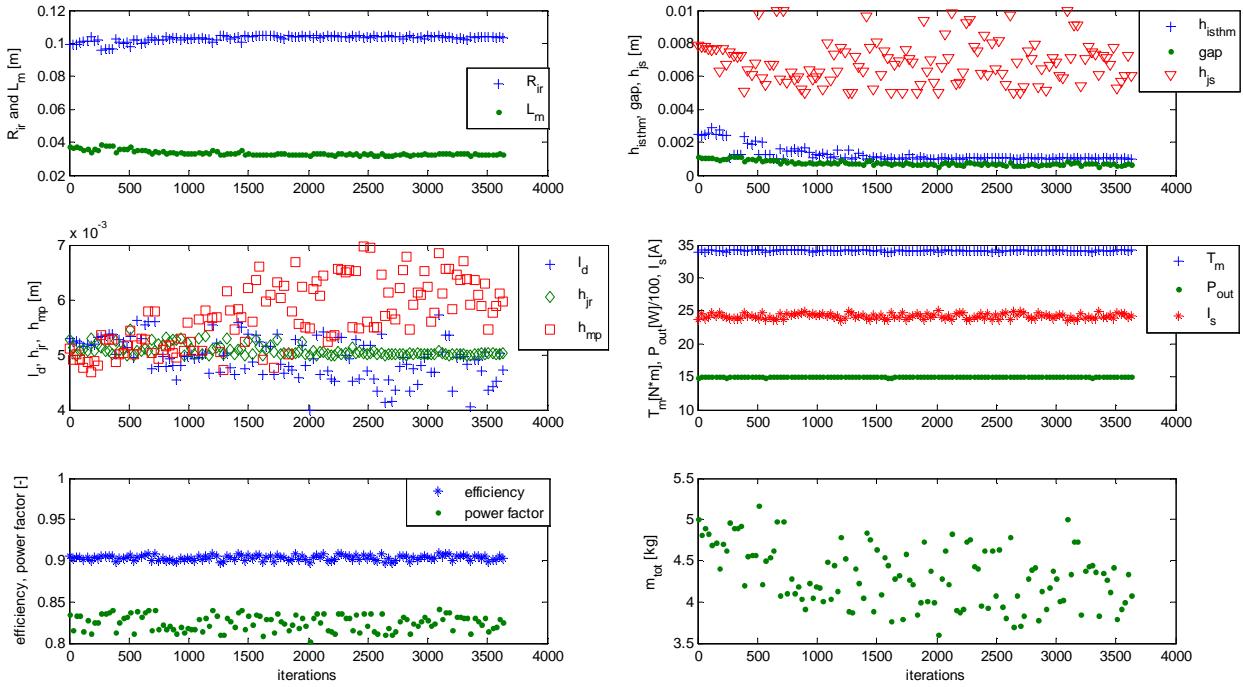


FIGURE 2. OR-PMSM optimization results.

The best optimized solution is presented against the original performances (obtained by the analytical model). An important improvement should be noticed, in terms of mass minimization, see Table VI.

TABLE VI: OR-PMSM Optimization Results

Symbol	After optimization	Gain (%)
m_{tot}	3.7697 kg	+ 47.0599
P_{fe}	37.7416 W	- 9.3151
P_{co}	118.9112 W	- 13.1911
η	0.9014	- 1.1622
$\cos\varphi$	0.8230	- 5.8244
c_{tot}	59.3902 €	+ 28.5823
Dir	210 mm	
h_{jr}	5 mm	
h_{is}	1 mm	
h_{js}	5 mm	
I_d	5 mm	
h_{gap}	0.5 mm	
h_{mp}	5 mm	
L_m	31.6 mm	

The proposed algorithm presented here offers a mass decrease of 47%. This is a very good result, in comparison with other research works found in the literature, for example [18] (for an interior permanent magnet synchronous machine, the mass has decreased up to 32%). Also, since the active part mass has significantly decreased, the total price of the machine has decreased too. The percentage of decrease of the cost is lower than for the mass, because the PMs height (and, finally, the mass) has increased a little bit (from 3 mm to 5 mm).

(In Figure 2 the reader can see that the best result in terms of mass decrease is 3.61 kg. Even if it is the best solution, this result was not sustained by the FEM analysis: the desired torque was not possible to be obtained. This proves that the analytical approach has its intrinsic limits, in terms of accurate saturation evaluation; this limitation is common for every analytical design algorithm.)

The reader's attention is now focused on the 'Gain' column from Table VI, to see the energetic performances of the optimized OR-PMSM. The copper, efficiency and power factor have practically kept their initial values – the power factor can be corrected through appropriate vector control (not discussed here). On the other hand, the iron loss has increased, after the optimization process. More details should be given here.

The iron loss is computed in the analytical model as the sum of eddy currents and hysteresis losses, calculated in the stator core (teeth and yoke), based on the following equation:

$$P_{Fe} = p_{hybst_d} + p_{hybst_j} + p_{eddy_d} + p_{eddy_j} \quad (6)$$

with $p_{hybst_d} = k_{hybst} \cdot p \cdot \Omega \cdot B_d^2 \cdot m_d$,

$$p_{hybst_j} = k_{hybst} \cdot p \cdot \Omega \cdot B_j^2 \cdot m_j,$$

$$p_{eddy_d} = k_{eddy} \cdot (p \cdot \Omega)^2 \cdot B_d^2 \cdot m_d,$$

$$p_{eddy_j} = k_{eddy} \cdot (p \cdot \Omega)^2 \cdot B_j^2 \cdot m_j$$

where k_{hybst} and k_{eddy} are the hysteresis and eddy currents loss coefficients, respectively (given for each type of material), B refers to the flux density and m to the mass, the subscript letters refer to stator teeth, 'd', and stator yoke, 'j', p represents the number of pair of poles and Ω is the angular speed (measured in rad/s).

From Eq. 6 it is obvious that the iron loss is proportional with the square of the flux density and mass of the stator core. Nevertheless, the optimized solution has poor iron loss, even if the mass of the machine has decreased significantly! This can be explained by the fact that the iron loss depends on the square of the flux density. Actually, the flux density maximal value, computed in the stator teeth and yoke, has increased in our analytical model because

the geometrical parameters have been reduced (towards their minimum limits in Table 4). The lack of our designing model (and of all existing analytical models) is that the flux density repartition cannot be exactly computed for each millimetre of the stator core. Here, the analytical model takes into account a maximum flux density in the teeth, in the isthmus and in the yoke, and based on these values will compute the iron loss. A correct computation of the flux density in all parts of the stator core can be produced thanks to the FEM analysis. The FEM analysis permits the correct computation of iron loss.

IV.3. FEM results of the OR-PMSM

Before the evaluation of iron loss through FEM, it is recalled that the FEM analysis is generally recognized to produce accurate results, very close to the real performances. Here, the Flux2D software has been used. The goal of this study is to validate the analytical approach and the optimization results for the electrical machine under study. In order to prove if the OR-PMSM was correctly designed/optimized one should consider the obtained torque and iron loss computation via FEM, Figure 3.

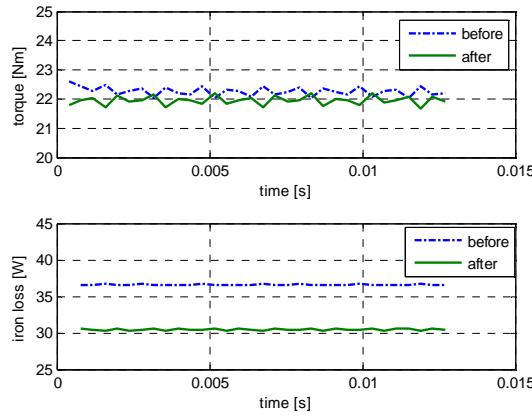


FIGURE 3. FEM results of the OR-PMSM, before and after optimization.

The rated current was injected as sinusoidal wave in the three phases winding, while rotor is running at rated speed. Thus, one can get the axis torque. By investigating Figure 3-top, it can be seen that the optimize solution provides a torque within the accepted limits (stated in Table IV). Also, one should observe that the torque variation is quite smooth. This is a very important improvement, especially for electrical machines designed for high torque/low speed applications. For example, the reluctance machines (like the one studied in [18]) present very high torque ripples. On the other hand, high number of poles and fractioned winding type machine offer very smooth torque wave forms. Of course, high number of poles involves high frequency, consequently higher iron loss.

From analytical design, the iron loss has been computed to approximately 34 W. After optimization of the mass of the machine, the iron loss has increased to 37 W. These values should be verified through FEM analysis, see Figure 3-bottom. While for the initial geometrical configuration (before optimization), the FEM analysis gave 36.5 W (very close to analytical value), after the optimization, the iron loss are of 31 W. The gain in *iron loss decrease* is of 15%, so, finally, the machines efficiency is better for the optimized variant! This is explained by the fact that the mass of the machine has drastically decreased (especially the length of the machine, being, after optimization, at 31 mm), and by the fact that the flux density repartition is not uniform in the stator core (some parts being saturated and others not – see the last comments from subsection III.C.1).

The numerical analysis has validated the optimization results, revealing also the weak points of the analytical model with regard to iron loss computation. Next, the study machine will be verified on test bench.

V. Experimental results

The goal of this section is to validate few optimized/numerical results of the designed machine. The operation of the realized OR-PMSM was verified on a test bench, Figure 4a. The tests were made in generator regime: in *no-load* conditions, *three-phase short-circuit* and *load* operation. The measured performances are presented in Figure 4b and Fig.5.

In Figure 4b-top it is plotted the electromotive force (*emf*) for different speed values. The FEM and test results are compared. A very good agreement was found (the error being below 4%). Also, the previous assumption related to the torque wave is sustained clearly here form Figure 4b-middle. Because of the fractioned type winding and high number of poles, the *emf* is practically a smooth sinusoidal; this means that the torque will present low ripples (as shown in Figure 3-top).

Another test was employed on the realized OR-PMSM. A 3 phase short circuit was produced, as usual for synchronous machines. A very low speed was considered, since, because of the high magnetic density of the PMs, the short-circuit current will increase rapidly. The value of 24 A, r.m.s., was rapidly attained, at 36 rpm. A really close result was obtained from tests and FEM, see Figure 4b-bottom, validating, again, our analytical and optimization approach.

In order to evaluate the energetic performances of the designed OR-PMSM, the load-generator operation has been employed. The mechanical power (the input power) needed to run the OR-PMSM is plotted in Fig. 5a at the bottom,

being the product of the angular speed (measured in rad/s) and the torque. The average values of the mechanical quantities are: 23.0306 N·m of torque, $649.8246 \text{ min}^{-1}$ (meaning 68.0495 rad/s) of speed and 1.5672e+003 W of mechanical power. As output, the OR-PMSM working in generator will provide 24.7505 of phase current, 26.7180 V of phase voltage and 1.4105e+003 W of electric power. This yields that the efficiency of the machine is 0.9 and the power factor is 0.79. Thus, very good experimental results were obtained, very close to the analytical-optimized performances (see Table VI for comparison).

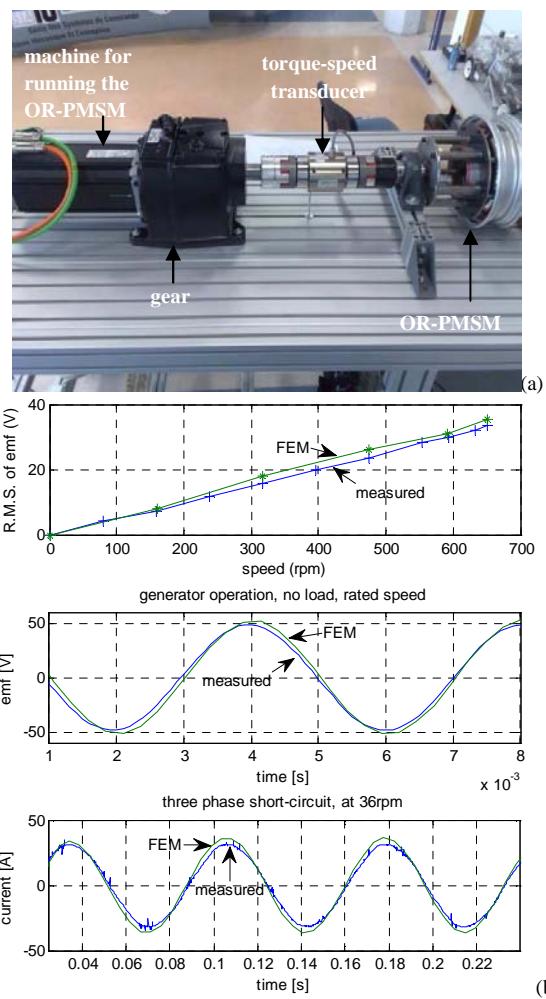


FIGURE 4. OR-PMSM: a) constructed motor; b) experimental results, in generator operation.

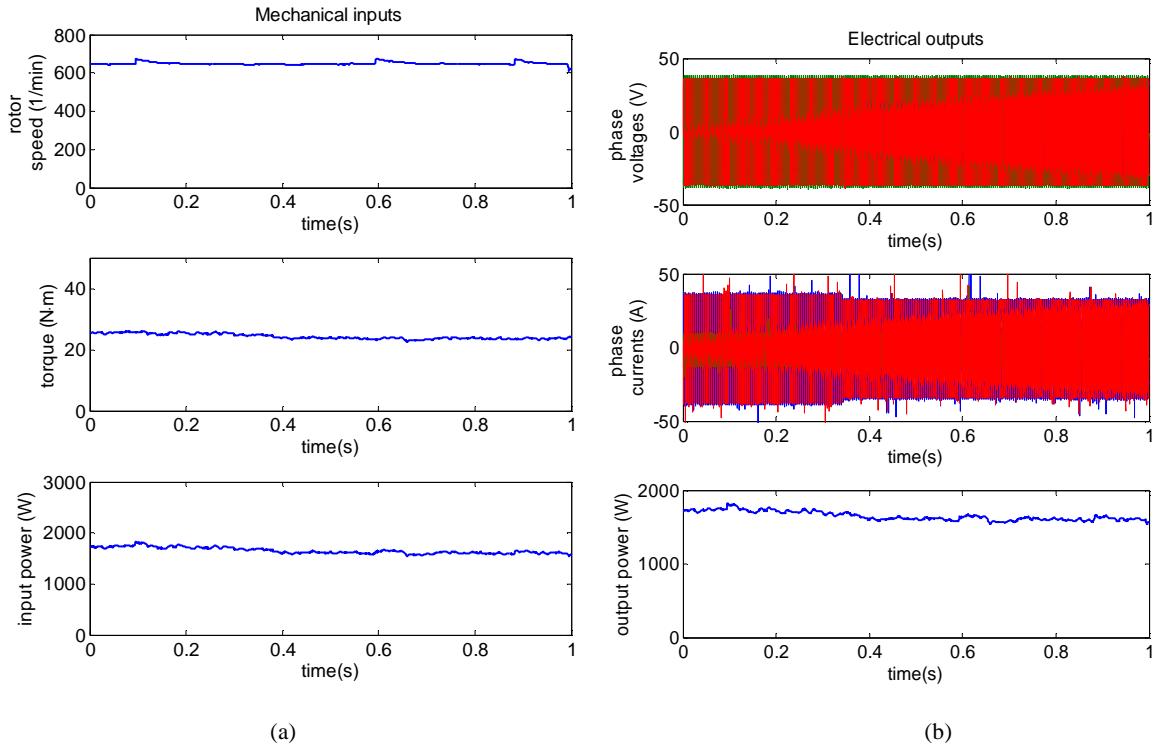


FIGURE 5. OR-PMSM working in generator regime: a) mechanical results; b) electrical results.

V. CONCLUSIONS

In this paper, it has been designed a novel Simulated Annealing algorithm for multi-objective optimization problem. We compared our SA approach with other algorithms, described in [3,5,15,16], on five well known benchmark functions. It has been shown that our approach has better performances in terms of accuracy, convergence rate, stability and robustness.

The SA approach was also employed for the optimization of a brushless synchronous machine, namely an outer-rotor permanent magnet synchronous machine (OR-PMSM). The objective function consists in the minimization of the total mass of the actives' parts of the machine. The obtained performances clearly show that the proposed SA algorithm overcomes other methods found in the scientific literature.

In future work, we will compare SA algorithm with hybrid algorithms (PSO-GA, PSO-SA) whose design is in progress by the authors. Comparison will also be done on additional benchmark functions and more complex problems including functions with dimensionality larger than 30.

REFERENCES

- [1] Al-khedhairi, A.: ‘Simulated Annealing Meta-heuristic for Solving P-Median Problem’. *Contemp. Math. Sciences*, Vol. 3, No.28, 2008, pp.1357-1365.
- [2] Olensek J., Tuma T., Puhan J., Bürmens A.: ‘A new asynchronous parallel global optimization method based on simulated annealing and differential evolution’. *Appl. Soft Comput.* Vol. 11, Issue 1, 2011, pp. 1481-1489.
- [3] Pant, M., Thangaraj, R., and Abraham, A.: ‘Particle Swarm Based Meta-Heuristics for Function Optimization and Engineering Applications’. *Proc. 7th Int. Conf. On Computer Information Systems and Industrial Management Applications*, Ostrava, Czech Republic, 2008, pp. 84 -90.
- [4] Leone C., Rogovich A., Marasini C., Genovesi S. and A. Monorchio A.: ‘Dynamic particle swarm optimisation for the design of loaded wire antennas’, *IET Microw. Antennas Propag.*, Vol. 5, Issue 5, 2011, p.611-615.
- [5] Olivetti de Franca, F., Palermo Coelho, G., Von Zuben, F. J., and R. de Faissol Attux, R.: ‘Multivariate ant colony optimization in continuous search spaces’. In *Proc. 10th Int. Conf. on Genetic and evolutionary computation*, Atlanta, GA, USA, 2008, pp. 9-16
- [6] Pei Luo, Huaipo Fu, Qiang Shi, Gang Sheng, Shanguo Huang, and Wanyi Gu: ‘Distributed routing based on improved ant colony algorithm in multi-domain WSON’, *Proc. of IET International Conference on Advanced Intelligence and Awareness Internet*, pp.95–98, 2010, Beijing, China.
- [7] Idoumghar L. and Schott R.: ‘Two Distributed Algorithms for the Frequency Assignment Problem in the Field of RadioBroadcasting’. *Journal of IEEE Transactions on Broadcasting*, Vol. 55, Issue 2, Part 1, 2009, pp. 223-229.
- [8] Gupta N., Swarnkar A., Niazi K.R., and Bansal R.C.: ‘Multi-objective reconfiguration of distribution systems using adaptive genetic algorithm in fuzzy framework’. *IET Gener. Transm. Distrib.* Vol. 4, Issue 12, 2010, pp. 1288–1298.
- [9] Dionísio R.P., Parca G., Reis C. and Teixeira A.L.: ‘Operational parameter optimisation of MZI-SOA using multi-objective genetic algorithms’, *IET, Electron. Lett.*, Vol 47, Issue 9, 2011, pp.561–562.
- [10] Hong Y.Y. and Chen Y.Y.: ‘Placement of power quality monitors using enhanced genetic algorithm and wavelet transform’, *IET Gener. Transm. Distrib.*, Vol 5, Issue 4, 2011, pp.461-466.
- [11] Michalewicz, Z., and Fogel, D. B.: ‘How to Solve It: Modern Heuristics’. Springer-Verlag Berlin Heidelberg New, 2000.

- [12] Hime A. Oliveira Jr., Antonio Petraglia: ‘Global optimization using dimensional jumping and fuzzy adaptive simulated annealing’. *Journal of Applied Soft Computing*, 2011, pp. 4175-4182.
- [13] Ramazan Şahin: ‘A simulated annealing algorithm for solving the bi-objective facility layout problem’. *Journal Expert Systems with Applications*, Vol. 38, Issue 4, 2011, pp. 4460-4465.
- [14] Atabak Elmia, Maghsud Solimanpurb, Seyda Topaloglu and Afshin Elmic: ‘A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts’. *Journal of Computers & Industrial Engineering*, Vol. 61, Issue 1, 2011, pp. 171-178.
- [15] Pourtakdoust, S. H., and Nobahari, H.: ‘An extension of ant colony system to continuous optimization problems’. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, and T. Sttzle, editors, ANTS Workshop, Vol. 3172 of Lecture Notes in Computer Science, Springer, 2004, pp.294-301.
- [16] Socha, K., and Dorigo, M.: ‘Ant colony optimization for continuous domains’. *Europ. Journal of Operational Research*, In Press, Corrected Proof, 2006.
- [17] Fodorean, D., Djerdir, A., Viorel, I.A., Miraoui, A.: ‘A Double Excited Synchronous Machine for Direct Drive Application - Design and Prototype Tests’. *IEEE Trans. on Energy Conversion*, Vol. 22, Issue 3, 2007, pp. 656-665.
- [18] Giurgea, S., Fodorean, D., Cirrincione, G., Miraoui, A., Cirrincione, M.: ‘Multimodel Optimization Based on the Response Surface of the Reduced FEM Simulation Model With Application to a PMSM’. *IEEE Trans. on Magnetics*, Vol. 44, Issue 9, 2008, pp. 2153-2157.
- [19] Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., and Chen, Y. P., Auger, A., and Tiwari, S.: ‘Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization’. Tech. Report 2005005, Nanyang Technological University, Singapore and IIT Kanpur, India, 2005.

Annexe : Quelques unes de mes publications

Metropolis Particle Swarm Optimization Algorithm with Mutation Operator For Global Optimization Problems

L. IDOUMGHAR^{*†}, M. IDRISI AOUAD[†], M. MELKEMI^{*} and R. SCHOTT[‡]

^{*}LMIA - MAGE, Université de Haute-Alsace. 4, rue des Frères Lumière, 68093 Mulhouse, France

Email: {lhassane.idoumghar, mahmoud.melkemi}@uha.fr

[†]INRIA Nancy - Grand Est / LORIA. 615, rue du Jardin Botanique, 54600 Villers-Lès-Nancy, France

Email: Maha.IdrissiAouad@inria.fr

[‡]IECN - LORIA, Nancy-Université, Université Henri Poincaré. 54506 Vandoeuvre-Lès-Nancy, France

Email: Rene.Schott@loria.fr

Abstract—When a local optimal solution is reached with classical Particle Swarm Optimization (PSO), all particles in the swarm gather around it, and escaping from this local optima becomes difficult. To avoid premature convergence of PSO, we present in this paper a novel variant of PSO algorithm, called MPSOM, that uses Metropolis equation to update local best solutions (lbest) of each particle and uses mutation operator to escape from local optima. The proposed MPSOM algorithm is validated on seven standard benchmark functions and used to solve the problem of reducing memory energy consumption in embedded systems (Scratch-Pad Memories SPMs). The numerical results show that our approach outperforms several recently published algorithms.

Keywords-Particles Swarm Optimization; Hybrid Algorithm; Global Optimization; Benchmark Functions; Scratch-Pad Memories;

I. INTRODUCTION

Global Optimization¹ (GO) is the branch of applied mathematics and numerical analysis that focuses on, well, optimization. GO can be defined as follows: *Minimize* $f(x) : S \rightarrow R$ where $f(\cdot)$ is the objective function (called also fitness value) which is subject to optimization, $S \subset \Re^D$ and D is the dimension of the search space S . Solving global optimization problem means that we need to find $x^* \in S$ such that $f(x^*) \leq f(x), \forall x \in S$. x^* is called the global minimizer of $f(\cdot)$ and $f(x^*)$ is called the global minimum value of $f(\cdot)$. Finding this global minimum is very difficult due to the existence of several local optima. Over the last decades, several optimization heuristics have been proposed for solving GO. Among the various heuristics let us mention Simulated Annealing (SA) [1], Genetic Algorithm [2] and optimization algorithms that make use of social or evolutionary behaviors like Particle Swarm Optimization (PSO) [3], [4], [5]. PSO is quite popular heuristic for solving complex optimization problems but this method has strengths and limitations principally premature convergence. To avoid premature convergence [6] of PSO, many works in the PSO community try to hybridize PSO with other

heuristic algorithms [4], [5], [7], [8], [9], [10], [11]. Particle Swarm Optimization (PSO) is based on the social behavior of individuals living together in groups. Each individual tries to improve itself by observing other group members and imitating the better ones. That way, the group members are performing an optimization procedure which is described in [12]. The performance of the algorithm depends on the way the particles (i.e., potential solutions to an optimization problem) move in the search space with a velocity that is updated iteratively. Large body of research in the field has been devoted to the analysis and proposal of different motion rules (see [4], [13], [14], [15] for recent accounts of PSO research).

In this paper, we present a new algorithm, called MPSOM, that makes full use of the exploration ability of PSO and incorporates a mutation operator and Metropolis rule to jump out from local optimum. MPSOM has been validated on seven benchmark functions [16] and compared with PSO algorithms variants described in [4], [7], [17]. We also use MPSOM algorithm to solve the problem of reducing memory energy consumption in embedded systems. The simulation results show that MPSOM outperforms the above mentioned algorithms. Hence MPSOM is a good alternative for dealing with complex numerical function optimization problems. This paper is organized as follows. Section 2 introduces briefly PSO algorithm. Section 3 is devoted to a precise detailed description of MPSOM. In Section 4, a series of numerical experiments regarding solution quality, convergence rate and robustness are conducted to show the superiority of our MPSOM algorithm. Discussion and results obtained about solving the problem of reducing memory energy consumption in embedded systems are presented in Section 5. Finally, conclusions and further research aspects are given in Section 6.

II. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [18], inspired by social behavior patterns

¹<http://www.it-weise.de/projects/book.pdf>

of organisms that live and interact within large groups. In particular, it incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior.

The idea of PSO algorithm is that particles move through the search space with velocities which are dynamically adjusted according to their historical behaviors. Therefore, the particles have the tendency to move towards the better search area over the course of search process. PSO algorithm starts with a group of random (or not) particles (solutions) and then searches for optima by updating each generation. Each particle is treated as a volume-less particle (a point) in the n -dimensional search space. The i^{th} particle is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$. At each generation, each particle is updated by following two *best* values:

- The first one is the best solution (fitness) it has achieved so far (the fitness value is also stored). This value is called *cbest*.
- Another best value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called *gbest*. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called *lbest*.

At each iteration, these two best values are combined to adjust the velocity along each dimension, and that velocity is then used to compute a new move for the particle. The portion of the adjustment to the velocity influenced by the individual's previous best position (*cbest*) is considered the *cognition* component, and the portion influenced by the best in the neighborhood (*lbest* or *gbest*) is the social component.

With the addition of the inertia weight factor, ω , by Shi and Eberhart [19] (for balancing the global and the local search), these equations are:

$$v_{ij} = \omega \times v_{ij} + c_1 \times rand \times (cbest_{ij} - x_{ij}) + c_2 \times rand \times (gbest_{ij} - x_{ij}) \quad (1)$$

$$x_{ij} = x_{ij} + v_{ij} \quad (2)$$

where *rand* is a random number independently generated within the range of $[0,1]$ and c_1 and c_2 are two learning factors which control the influence of the social and cognitive components (usually, $c_1 = c_2 = 2$).

In Equation 1, if the sum on the right side exceeds some fixed V_{max} value, then the velocity on that dimension is assigned to be $\pm V_{max}$. Thus, particles' velocities are clamped to the range of $[-V_{max}; V_{max}]$ which serves as a constraint to control the global exploration ability of PSO algorithm. This also reduces the likelihood of particles for leaving the search space. Note that this does not restrict the values of x_i to the range $[-V_{max}; V_{max}]$; it only limits the maximum distance that a particle will move during one iteration.

```

Initialize swarm_size particles
Evaluate(Swarm)
stop_criterion←maximum evaluation functions
noImprove ← 0,
Initialize inertia factor  $w \leftarrow w_0$ 
Initialize Initial Temperature  $T \leftarrow T_0$ 
while Not stop_criterion do
    Sort(Swarm)
    if nonimprove <  $k$  then
        for each particle  $i \leftarrow 1$  to swarm_size do
            Accept( $X$ , cbest,  $T$ )
            Update velocity according to Equation (4)
            Enforce velocity bounds
            Update particle position according to Equation (2)
            Enforce position bounds
        end
    else
        // Mutation operator
        noImprove ← 0
        for each particle  $i \leftarrow 1$  to swarm_size do
            Initialize its velocity to maximum velocity allowed
        end
    end
    Evaluate(Swarm)
    if there is no improvement of global best solution then
        | noImprove ← noImprove + 1
    else
        | Update global best solution
        | noImprove ← 0
    end
    Update inertia factor  $w$  by using Equation 6
    Update ( $T$ )
    Update (stop_criterion)
end

```

Algorithm 1: MPSOM Algorithm.

III. MPSOM ALGORITHM

This section presents MPSOM algorithm that makes full use of the exploration ability of PSO and incorporates a mutation operator and Metropolis rule to jump out from local optimum. The mutation operator is introduced to PSO every K iterations if no improvement of the global best solution occur. The value of K is predefined to 60 according to our experimentations.

The principle of MPSO algorithm works as illustrated in Algorithm 1, where:

- **Description of a particle:** Each particle (solution) is represented by its:
 - current position $X \in S$ represented by $n > 0$ components, i.e., $X = (x_1, x_2, \dots, x_n)$, where $i = 1, 2, \dots, n$ and n represents the dimension of the optimization problem to solve.
 - previous best solution founded so far (called *cbest*).
 - velocity that corresponds to the rate of the position change.

- **Initial Swarm:** Initial Swarm corresponds to population of particles that will evolve. Each particle x_i is initialized with uniform random value between the lower and upper boundaries of the interval defining the optimization problem.
- **Evaluate function:** Evaluate (or fitness) function in MSOSM algorithm is typically the objective function that we want to minimize in the problem.
- **Sort:** All particles of the swarm are sorted in decreasing order of their objective function.
- **Accept function:** Function $Accept(cbest, X, T)$ is decided by the acceptance probability given by Equation 3, which is the probability of accepting current position of one particle as its $cbest$.

The probability of accepting a new solution is given as follows:

$$p = \begin{cases} 1 & f(X) \leq f(cbest) \text{ or} \\ & rand \times (1 + e^{\frac{f(X)-f(cbest)}{T}}) < 2.0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where:

- T is the temperature. It plays a similar role as the temperature in the physical annealing process [1].
- $rand$ is a random number independently generated within the range of $[0, 1]$.

- **Update velocity:** We propose in this work a topological neighborhood which is determined dynamically during the search process, according to the objective function of the swarm. After we had sorted the Swarm population in decreasing order of their objective function, for each i^{th} particle, a set N_i of its neighbors is defined as: $N_i = \{particle k / k \geq i \text{ and } k \leq swarm_size\}$. Then, to adjust velocity of i^{th} particle, we used the following equation:

$$v_{ij} = \omega \times v_{ij} + c_1 \times rand \times (cbest_{ij} - x_{ij}) + min(V_{max}, \sum_{k=i}^{swarm_size} \frac{cbest_{kj} - x_{ij}}{k}) \quad (4)$$

In Equation 4, the social component of i^{th} particle is computed as a weighted average of all best particles in N_i . In so doing, a particle is not only influenced by the global best solution. The particle adjusts its speed according to the weighted average of all solutions which are better than its ones. This means simply that it is better for a particle to follow a group of particles rather than a single one (this is of course common

sense).

- **Temperature update:** To avoid our algorithm getting trapped in a local minimum point, the rate of reduction should be slow. In this work, the following method to reduce the temperature has been used:

$$T_{i+1} = \gamma T_i \quad (5)$$

where $i = 0, 1, \dots$ and $\gamma = 0.99$.

Thus, at the start of MPSOM most worsening moves may be accepted, but at the end only improving ones are likely to be allowed. This can help the procedure jump out of a local minimum.

- **Update Inertial factor:** The inertial factor, used to control relation speed between previous and current speed of each particle, is defined as follows:

$$w_k = w_0 \times \left(1 - \frac{T_0 - T_k}{T_0}\right) \quad (6)$$

where:

- $w_0 = 0.9$ corresponds to the start weight value of the algorithm. Note that with lower value of w the searching region of the algorithm could be around the best solution, and with higher level of w , the algorithm could enhance the exploration searching (global searching).
- T_0 is an initial temperature and T_k represents temperature at the k^{th} iteration.

- **Mutation operator:** If no improvement of the global best solution occur during the last K iterations, then it means that the algorithm is trapped in a local optimum point. To escape out from local optimum, our MPSOM algorithm uses mutation operator based on the following idea: by giving a maximum velocity allowed to each particle, Equation 2 ensures that all particles will jump out from local optimum point and MPSOM algorithm could have a large range of exploration ability.

IV. EXPERIMENTS RESULTS ON BENCHMARK FUNCTIONS

To compare the performance of our MPSOM algorithm with those described in [7], [4], [17], we use seven benchmark functions [16] described in Table I. These functions possess some properties similar to real world problems and provide a good launch pad for testing the credibility of an optimization algorithm. For these functions, there are many local optima in their solution spaces. The amount of local optima increases with increasing complexity of the functions, i.e. with increasing dimension.

Table I
STANDARD BENCHMARK FUNCTIONS ADOPTED IN THIS WORK.

Function	Problem	Range	$f(x^*)$	ϵ	Classification
Sphere	$\sum_{i=1}^n x_i^2$	[-100;100]	0	0.01	Unimodal
Rastrigin	$\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[-5.12;5.12]	0	10	Multimodal
Griewank	$\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600; 600]	0	0.1	Multimodal
Rosenbrock	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[-2.048;2.048]	0	100	Unimodal
Schwefel	$420.9687 n - \sum_{i=1}^n (x_i \sin(\sqrt{ x_i }))$	[-500.0;500.0]	0	2000	Multimodal
Ackley	$20 + e - 20 e^{-0.2 (\frac{1}{n} \sum_{i=1}^n x_i^2)^{\frac{1}{2}}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)}$	[-30.0; 30.0]	0	0.1	Multimodal
Quartic	$(\sum_{i=1}^n i x_i^4) + rand[0, 1]$	[-1.28; 1.28]	0	0.1	Noisy

Comparison with results obtained by [4], [7], [17]

In order to make a fair comparison of classical PSO, Gaussian PSO with jumps algorithm (GPSO-J) [17], Comprehensive Learning PSO algorithm (CLPSO) [7], Particle Swarm Optimizer with Adaptive Tabu and Mutation (ATM-PSO) [4] and our MPSOM approach, we fixed, as indicated in [4], the number of particles in the swarm is 20 and the maximum number of function evaluations (FEs) is set to 5000 $D = 150000$. A total of 30 runs for each experimental setting were conducted and the average fitness of the best solutions throughout the run is recorded.

Analysis of numerical results² given in Table II show that:

- All the algorithms successfully solve Sphere's function which is the simplest unimodal function even if our algorithm obtains best solutions.
- Our MPSOM algorithm outperforms other PSO variants when optimizing hard multimodal problems. In fact, for Rosenbrock's function, which is considered in the optimization literature as difficult problem due to the nonlinear interaction between variables [21], we can see that our algorithm successfully solves this problem while the other algorithms get stuck in local optima.
- ATM-PSO and CLPSO successfully solve Schwefels problem. In contrast, GPSO-J and our MPSOM algorithm fail to optimize this problem.
- For highly multimodal Rastrigins function, ATM-PSO and our MPSOM approach hit the global optimal

solution reliably within 150000 FEs.

- The analysis of the results obtained for Ackley's function shows that ATM-PSO and MPSOM obtain better mean results than GPSO-J and CLPSO.
- The optimization of the Griewanks function is an example of the failure of all algorithms except MPSOM that hits the global optimal solution.
- The optimization of the Quartic noisy function is another example of the success of MPSOM and illustrates that other algorithms are adapted to a dynamic environment.

Convergence Rate: $Q_{measure}$

In order to evaluate the convergence rate of all algorithms compared in this section, a threshold ϵ is set for each benchmark function as described in [4]. When each algorithm reaches the specified threshold ϵ (see Table I) of a certain benchmark function in the k^{th} trial, the number of function evaluations FE_k needed is recorded and the current trial k is denoted as a successful trial.

[22] proposes a $Q_{measure}$ criterion which incorporates the measure of both convergence and robustness. The $Q_{measure}$ [4] used, to evaluate performance of all algorithms, is defined as follows:

$$Q_{measure} = \frac{n_t \sum_{i=1}^{n_s} FE_i}{n_s^2} \quad (7)$$

where:

- n_t denotes the total number of trials,
- n_s denotes the number of successful trials,
- the success ratio SR is defined as $SR = \frac{n_s}{n_t}$.

²Mean Results of MPSOM can be downloaded from [20]

Table II
MEAN RESULTS AND STANDARD DEVIATION OVER 30 INDEPENDENT RUNS FOR 4 PSO VARIANTS ALGORITHMS ON SEVEN BENCHMARK FUNCTIONS (DIMENSION $n = 30$ AND SWARM SIZE = 20).

Function	<i>CLPSO</i>	<i>GPSO – J</i>	<i>ATM – PSO</i>	MPSOM
Sphere	8.99e-14	3.85e-07	8.90e-104	1.17e-114
	$\pm 4.66e-14$	$\pm 7.97e-07$	$\pm 3.18e-103$	$\pm 1.17e-114$
Rosenbrock	20.88	27.11	15.13	1.44e-02
	± 2.58	± 16.91	$\pm 8.71e-01$	$\pm 1.63e-02$
Schwefel	1.76e-12	1336.29	2.36e-12	8.11e-01
	$\pm 3.27e-13$	± 290.81	$\pm 9.57e-13$	$\pm 8.59e-01$
Rastrigin	1.34e-06	15.86	0	0
	$\pm 1.66e-06$	± 4.52	± 0	± 0
Ackley	8.45e-08	1.43e-03	2.59e-14	1.16e-10
	$\pm 1.96e-08$	$\pm 1.36e-03$	$\pm 6.12e-15$	$\pm 2.42e-11$
Griewank	1.95e-09	4.02e-02	2.22e-02	0
	$\pm 4.35e-09$	$\pm 4.02e-02$	$\pm 2.03e-02$	± 0
Quartic	8.18e-03	4.45e-03	9.77e-03	1.92e-05
	$\pm 2.39e-03$	$\pm 9.95e-04$	$\pm 3.16e-03$	$\pm 2.84e-05$

Table III
 $Q_{measure}$ VALUES OF FOUR ALGORITHMS ON SEVEN BENCHMARK FUNCTIONS STUDIED IN THIS WORK. PERCENTAGE IN PARENTHESES ARE SUCCESS RATIO (SR).

Function	<i>CLPSO</i>	<i>GPSO – J</i>	<i>ATM – PSO</i>	MPSOM
Sphere	67977 (100%)	19343 (100%)	8101 (100%)	539 (100%)
	34654 (100%)	6537 (100%)	980 (100%)	932 (100%)
Rosenbrock	24439 (100%)	47987 (100%)	9726 (100%)	22821 (100%)
	87201 (100%)	1795500 (7%)	15187 (100%)	3597 (100%)
Schwefel	62659 (100%)	42857 (100%)	9664 (100%)	5963 (100%)
	65437 (100%)	18309 (90%)	7140 (100%)	6185 (100%)
Rastrigin	38094 (100%)	11137 (100%)	5988 (100%)	1223 (100%)

$Q_{measure}$ ³ values of four algorithms on 7 benchmark functions studied in this work are summarized in Table III. Analysis of the table shows that MPSOM converges slightly faster than all other algorithms, except for the optimization of Schwefel's function where CLPSO and ATM-PSO achieve very good performance in comparison with our MPSOM.

V. REDUCING MEMORY ENERGY CONSUMPTION IN EMBEDDED SYSTEMS

A. Description of the SPM Problem

Reducing memory energy consumption of embedded systems is crucial. To do so, we focus on software techniques working on the memory management. Most authors rely on Scratch-Pad Memories (*SPMs*) rather than caches [24]. Although cache memory helps a lot with program speed, it is not the most appropriate for embedded systems. In fact, cache increases the system size and its energy cost (cache area plus managing logic). Like cache, SPM consists

of small, fast SRAM. The main difference is that SPM is directly and explicitly managed at the software level, either by the developer or by the compiler which makes it more predictable. SPM requires up to 40% less energy and 34% less area than cache [25]. In this paper, we will therefore use an SPM in our memory architecture. Due to the reduced SPM size, we allocate interesting data in it whereas the other data are placed in main memory (*DRAM*). In order to determine interesting data, we use *data profiling* to gather memory access frequency information.

B. Optimization Problem

Problem of reducing memory energy consumption is a kind of knapsack problem [26]. We want to fill SPM that can hold a maximum capacity of C with some combination of data from a list of N possible data each with $size_i$ and $access\ number_i$ so that the access number of the data allocated into SPM is maximized. If N is the total number of data, then a solution is just a finite sequence s of N terms such that $s[n]$ is either 0 or the size of the n_{th} data. $s[n] = 0$ iff the n_{th} data is not selected in the solution. This solution

³ $Q_{measure}$ of MPSOM can be downloaded from [23]

must satisfy the constraint of not exceeding the maximum SPM capacity (i.e. $\sum_{i=1}^N s[i] \leq C$).

C. Discrete MPSOM

In order to reduce memory energy consumption, some features of Algorithm 1 must be adapted:

- **Solution (particle):** a solution can be represented by an array having size equal to the number of the data. Each element from this array denotes whether a data is included in the SPM ('1') or not ('0'). The MPSOM algorithm starts with an initial swarm which is randomly initialized.
- **Evaluate function:** is the objective function that we want to minimize in the problem. It serves for each solution to be tested for suitability to the environment under consideration.

$$\text{Evaluate}(\text{solution}) = \text{Total_Number_Access_all_data} - \text{Number_Access}(\text{solution}). \quad (8)$$

- **Position Update Equation:** each dimension j of the particle i is updated by using Equation 9.

$$x_{ij} = \begin{cases} 1 & \text{rand} < \text{sigm}(v_{ij}) \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where $\text{sigm}(v_{ij})$ is the sigmoid function, used to scale the velocities between 0 and 1, defined as:

$$\text{sigm}(v_{ij}) = \frac{1}{1 + \exp(-v_{ij})} \quad (10)$$

D. Memory Energy Estimation Model

In order to compute the energy cost of the system, we propose in this section an energy consumption estimation model (explained in [27]) for our considered memory architecture composed by an SPM, an instruction cache and a DRAM. Equation 11 gives the energy model where the three terms refer to the total energy consumed respectively in SPM, in instruction cache and in DRAM.

$$E = E_{tspm} + E_{tic} + E_{tdram} \quad (11)$$

In this model, we distinguish between the two cache write policies: Write-Through (WT) and Write-Back (WB). In a WT cache, every write to the cache causes a synchronous write to DRAM. Alternatively, in a WB cache, writes are not immediately mirrored to DRAM. Instead, the cache tracks which of its locations have been written over and then, it marks these locations as dirty. The data in these locations is written back to DRAM when those data are evicted from the cache [28]. Our aim is to minimize the detailed energy estimation model presented below:

$$E = N_{spmr} * E_{spmr} \quad (12)$$

$$+ N_{spmw} * E_{spmw} \quad (13)$$

$$+ \sum_{k=1}^{N_{icr}} [h_{ik} * E_{icr} + (1 - h_{ik}) * [E_{dramr} + E_{icw} \\ + (1 - WP_i) * DB_{ik} * (E_{icr} + E_{dramw})]] \quad (14)$$

$$+ \sum_{k=1}^{N_{icw}} [WP_i * E_{dramw} + h_{ik} * E_{icw} + (1 - WP_i) * \\ (1 - h_{ik}) * [E_{icw} + DB_{ik} * (E_{icr} + E_{dramw})]] \quad (15)$$

$$+ N_{dramr} * E_{dramr} \quad (16)$$

$$+ N_{dramw} * E_{dramw} \quad (17)$$

Lines (12) and (13) represent respectively the total energy consumed during a reading and during a writing from/into SPM. Lines (14) and (15) represent respectively the total energy consumed during a reading and during a writing from/into instruction cache. When, lines (16) and (17) represent respectively the total energy consumed during a reading and during a writing from/into DRAM. The various terms used in this energy model are explained in Table IV.

Table IV
LIST OF TERMS.

Term	Meaning
E_{spmr}	Energy consumed during a reading from SPM.
E_{spmw}	Energy consumed during a writing into SPM.
N_{spmr}	Reading access number to SPM.
N_{spmw}	Writing access number to SPM.
E_{icr}	Energy consumed during a reading from instruction cache.
E_{icw}	Energy consumed during a writing into instruction cache.
N_{icr}	Reading access number to instruction cache.
N_{icw}	Writing access number to instruction cache.
E_{dramr}	Energy consumed during a reading from DRAM.
E_{dramw}	Energy consumed during a writing into DRAM.
N_{dramr}	Reading access number to DRAM.
N_{dramw}	Writing access number to DRAM.
WP_i	The considered cache write policy: WT or WB. In case of WT, $WP_i = 1$ else in case of WB then $WP_i = 0$.
DB_{ik}	Dirty Bit used in case of WB to indicate during the access k if the instruction cache line has been modified before ($DB_i = 1$) or not ($DB_i = 0$).
h_{ik}	Type of the access k to the instruction cache. In case of cache hit, $h_{ik} = 1$. In case of cache miss, $h_{ik} = 0$.

E. Experimental Results

To compare the performance of our algorithm with the Tabu Search Algorithm (TS) described in [27], we use eleven benchmarks from six different suites: MiBench [24], SNU-RT, Mlardaen, Mediabenchs, Spec 2000 and Wcet Benchs. Table V gives a description of these benchmarks and they also can be downloaded from [29]. Table V gives a description of these benchmarks. In experiments, 30 different executions for each heuristic are performed and the best and average results obtained on these 30 executions are recorded. In this case, the best and the average solutions give similar

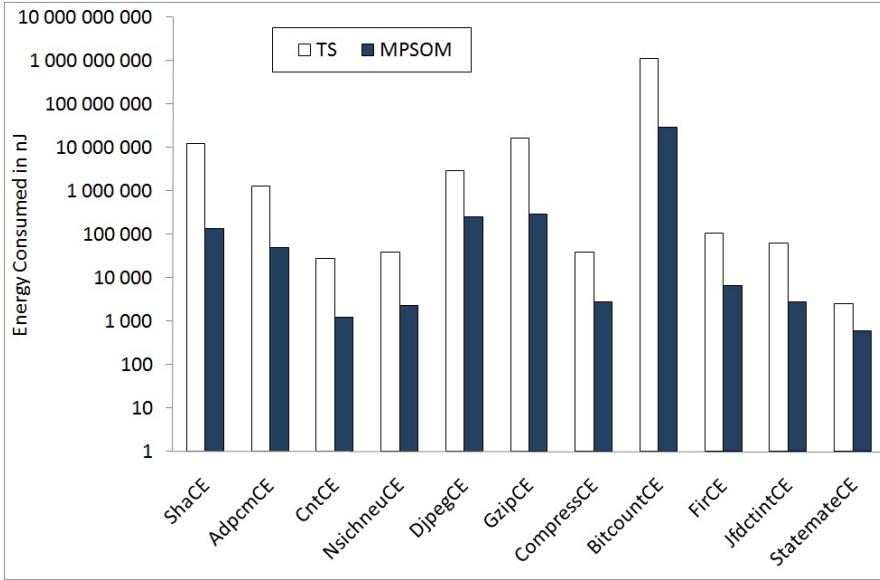


Figure 1. Energy consumed by our benchmarks.

Table V
LIST OF BENCHMARKS.

Benchmark	Suite	Description
ShaCE	MiBench	The secure hash algorithm that produces a 160-bit message digest for a given input.
BitcountCE	MiBench	Tests the bit manipulation abilities of a processor by counting the number of bits in an array of integers.
FirCE	SNU-RT	Finite impulse response filter (signal processing algorithms) over a 700 items long sample.
JfdctintCE	SNU-RT	Discrete-cosine transformation on 8x8 pixel block.
AdpcmCE	Mälardalen	Adaptive pulse code modulation algorithm.
CntCE	Mälardalen	Counts non-negative numbers in a matrix.
CompressCE	Mälardalen	Data compression using lzw.
DjpegCE	Mediabenchs	JPEG decoding.
GzipCE	Spec 2000	Compression.
NsichneuCE	Wcet Benchs	Simulate an extended Petri net. Automatically generated code with more than 250 if-statements.
StatemateCE	Wcet Benchs	Automatically generated code.

results. We therefore call MPSOM the common value of their executions. Figure 1 shows that our MPSOM algorithm outperforms TS. In fact, our MPSOM algorithm consumes from 76.23% (StatemateCE) up to 98.92% (ShaCE) less energy than TS.

VI. CONCLUSION

In this paper, we have designed a new algorithm (MPSOM) that uses the exploration ability of PSO and mutation operator to avoid premature convergence. Compared with

ATM-PSO [4], CLPSO [7] and GPSO-J [17] on seven well-known benchmark functions (unimodal, noisy, multimodal) and on the problem of reducing memory energy consumption in embedded systems, it has been shown that our approach has better performances in terms of accuracy, convergence rate, stability and robustness. In future work, we will compare MPSOM algorithm with other hybrid algorithms (GASA, PSO-GA) whose design is in progress. Comparison will also be done on additional benchmark functions and more complex problems including functions with dimensionality larger than 30.

ACKNOWLEDGMENT

The work of M. Idrissi Aouad is supported by the french national research agency (ANR) in the Future Architectures program.

REFERENCES

- [1] M. Locatelli. *Simulated annealing algorithms for continuous global optimization*. In: P.M. Pardalos, H.E. Romeijn (Eds.) Handbook of Global Optimization, Vol. 2, pp. 179-230, Kluwer Academic Pub. 2002.
- [2] L. Idoumghar and R. Schott. *Two Distributed Algorithms for the Frequency Assignment Problem in the Field of RadioBroadcasting*, Journal of IEEE Transactions on Broadcasting, Vol. 55, Issue 2, Part 1, pp. 223-229, June 2009.
- [3] M. Pant, R. Thangaraj and A. Abraham. *Particle Swarm Based Meta-Heuristics for Function Optimization and Engineering Applications*. 7th Conf. Computer Information Systems and Industrial Management Applications. Vol. 7, pp. 84-90, Issue , 26-28, 2008.

- [4] W. Yu-Xuan, X. Qiao-Liang, Z. Zhen-Dong. *Particle swarm optimizer with adaptive tabu and mutation: A unified framework for efficient mutation operators*, Journal ACM Transactions on Autonomous and Adaptive Systems (TAAS), Vol. 5 , Issue 1, February 2010.
- [5] K. Premalatha and A. M. Natarajan. *Combined Heuristic Optimization Techniques for Global Minimization* Journal of Advances in Soft Computing and Its Applications, Vol. 2, No. 1, pp. 85-99, March 2010.
- [6] A. Ratnaweera, S. Halgamuge and H. Andwatson. *Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients*, IEEE Trans. Evol. Comput. Vol. 8, Issue 3, pp. 240255. 2004.
- [7] J. J. Liang, A. K. Qin, P. N. Suganthan and S. Baskar. *Comprehensive learning particle swarm optimizer for global optimization of multimodal functions*, IEEE Trans. Evol. Comput. Vol. 10, N. 3, pp. 281295, 2006.
- [8] Jui-Fang Chang, Shu-Chuan Chu, John F. Roddick and Jeng-Shyang Pan, *A Parallel Particle Swarm Optimization Algorithm with Communication Strategies*, Journal of Information Science and Engineering, Vol. 21, No. 4, pp. 809-818, 2005.
- [9] Chao-Li Sun, Jian-Chao and Jeng-Shyang Pan, *An Improved Particle Swarm Optimization with Feasibility-Based Rules for Constrained Optimization Problems*, IEA/AIE 2009, pp. 202-211, 2009.
- [10] S-K. S. Fan and E. Zahara. A hybrid simplex search and particle swarm optimization for unconstrained optimization. European Journal of Operational Research, Vol. 181, N. 2, pp. 527-548, 2007.
- [11] Y-T. Kao and E. Zahara. A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Applied Soft Computing, Vol. 8, N. 2, pp. 849-857, 2008.
- [12] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Academic Press, 2001.
- [13] M. A. Montes de Oca and T. Stützle. *Convergence behavior of the fully informed particle swarm optimization algorithm*. Proceedings of the 10th annual conference on Genetic and evolutionary computation. pp. 71-78, 2008.
- [14] A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, Chichester, West Sussex, England, 2005.
- [15] R. Poli, J. Kennedy and T. Blackwell. *Particle swarm optimization. An overview*. Swarm Intelligence, 1(1):3357, 2007.
- [16] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. *Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization*. Technical Report 2005005, Nanyang Technological University, Singapore and IIT Kanpur, India, May 2005.
- [17] R. A. Krohling. *Gaussian particle swarm with jumps*. Proceedings of the IEEE Congress on Evolutionary Computat. pp. 12261231, 2005.
- [18] J. Kennedy and R. C. Eberhart. *Particle swarm optimization*. In Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.
- [19] Y. Shi and R. C. Eberhart. *A modified particle swarm optimizer*. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1998), Piscataway, NJ. pp. 69-73, 1998
- [20] www.lmia.uha.fr/~mage/idoumghar/MPSOMMeansSTD.zip
- [21] D. Ortiz-Boyer, C. Herbas-Martínez, and N. Garcíá-Pedrajas. *CIXL2 - A crossover operator for evolutionary algorithms based on population features*. Journal of Artificial Intelligence Research, vol.24: pp. 1-48, 2005.
- [22] Feoktistov. *Differential Evolution: In Search of Solutions*, Vol. 5, 2006, Springer.
- [23] <http://www.lmia.uha.fr/~mage/idoumghar/MPSOMQmesure.zip>
- [24] M. Idrissi Aouad and O. Zendra. *A Survey of Scratch-Pad Memory Management Techniques for low-power and -energy*, Proc. of International Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOLPS'07), 2007.
- [25] H.Ben Fradj, A. El Ouardighi, C. Belleudy and M. Auquin. *Energy aware memory architecture configuration*, ACM SIGARCH Computer Architecture News, Vol. 33 , Issue 3, pp. 3 - 9, 2005.
- [26] H. Kellerer, U. Pferschy and D. Pisinger. *Knapsack Problems*, Springer, Berlin, Germany, 2004.
- [27] M. Idrissi Aouad, R. Schott and O. Zendra. *A Tabu Search Heuristic for Scratch-Pad Memory Management*. Proc. of International Conference on Software Engineering and Technology (ICSET 2010), Vol. 64, pp. 386-390, 2010.
- [28] A. Tanenbaum. *Architecture de l'ordinateur 5e édition*, Pearson Education, November, 2005.
- [29] Benchmarks, www.loria.fr/~idrissma/benchs.zip
- [30] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge and R. B. Brown. *MiBench: A free, Commercially Representative Embedded Benchmark Suite*, WWC '01: Proceedings of the Workload Characterization, pp. 3-14, Washington, DC, USA, 2001.

Pattern retrieval from a cloud of points using geometric concepts

L. Idoumghar and M. Melkemi

Université Haute Alsace, Laboratoire de Mathématiques Informatique et Applications. 4 rue des frères Lumière, 68093, Mulhouse, France.
lhassane.idoumghar@uha.fr, mahmoud.melkemi@uha.fr

Abstract. In this article, we propose a generalization of the Euclidean α -shape, and through it we present an algorithm to retrieval shapes that are similar to a query shape from a finite set of points. Similarity means that we look for patterns identical, according to a given measure, to a query shape independently of translation, rotation and scaling transforms.

1 Introduction

In this article, we are interested in the problem of detecting patterns embedded in a cloud of points and similar to a query shape. In contrast with existing approaches [1], [3], [4], our method uses only geometric algorithms and treats cloud of points. According to a defined similarity measure, the algorithm detects accurately the overall patterns similar to the query shape. It is based on a mathematical formalism that we considered it as a generalization of the α -shape concept [2] and it is implemented using the Voronoi diagram [5].

In our approach, the query shape is defined by a union of discs arranged according to a fixed configuration. It depends only on three parameters, two points and a positive real number, they, respectively, allow to move and to dilate/shrink the query shape, some examples are shown in Figs. 1, 2 and 3. Given a query shape represented by a union of discs and a set of points, the presented algorithm computes the gaps embedded in the set of points and similar to the query shape.

This article is organized as follows: first, we present geometrical concepts related to the concept we introduce and its implementation. In section 3 and 4 we present the main concepts we introduce; the query shape definition, the α -shape generalization and the algorithms.

2 Related geometrical concepts

Let $P = \{p_1, p_2, \dots, p_n\}$ be a set of n distinct points of the plan. In this paper we assume that the points of S are in general-position. This means that no four points lie on a common circle. The general-position assumption simplifies the forthcoming definitions. Even if the points are not in general position, there

exists many efficient algorithms to compute the Voronoi diagram of a set of points [5].

2.1 Definition of the Voronoi diagram

Let pq be the Euclidean distance between two points p and q . Usually, we define the Voronoi polygon $R(p_i)$ of p_i as the set of points $p \in \mathbb{R}^2$ so that $pp_i \leq pp_j$, for all $p_j \in P$. The Voronoi diagram of P is the set $V(P) = \{R(p_1), \dots, R(p_n)\}$. The dual of the Voronoi diagram, obtained by connecting the points p_i to their neighbors in $V(P)$, is the Delaunay triangulation of P and noted $D(P)$. To emphasize the morphological aspect of the Voronoi diagram, we rewrite the definition of $R(p_i)$ as follows: let $x \in \mathbb{R}^2$ and $p \in \mathbb{R}^2$ be two points and r a positive real number. We note $b_p(x)$ an open disc centered at p and the point x is on its boundary ($x \in \partial b_p(x)$, where ∂S denotes the boundary of a region S). The definition of $R(Sp_i)$ can be generated using discs $b_p(p_i)$. The polygon $R(p_i)$ corresponds to the set: $R(p_i) = \{p \in \mathbb{R}^2 \mid b_p(p_i) \cap P = \emptyset\}$.

2.2 The Euclidean α -shape of a point set

Let b_α be an open disc, of radius α . b_α is said to be *empty* if and only if $b_\alpha \cap P = \emptyset$. The α -shape of P , we note $F_\alpha(P)$, is the set of the edges $[p_i p_j]$ such that there exists an empty disc b_α , where $\{p_i, p_j\} \subset \partial b_\alpha$.

The α -shape is a sub-graph of the Delaunay triangulation. An edge $s = [p_i p_j]$ of $D(P)$ is an element of $F_\alpha(P)$ if and only if $\alpha_{min}(s) \leq \alpha \leq \alpha_{max}(s)$, where $\alpha_{min}(s)$ and $\alpha_{max}(s)$ are two positive real numbers computed from the edges of $V(P)$. Indeed, the $\alpha_{min}(s)$ is the minimum distance of p_i to the Voronoi edge separating p_i from p_j , the $\alpha_{max}(s)$ corresponds to the maximum distance (Edelsbrunner and Müke, 1994). In the next section, we show how to represent a query-shape by a union of discs.

3 The query-shape definition

3.1 A unitary query-shape

Consider a set of points P and denote $b(c, r)$ an open disc of radius r and centered at c . Let S_1^m be a connected planar graph, its set of m vertices is denoted C_1^m . We associate to each vertex $c \in C_1^m$ a weight $c' > 0$. The graph S_1^m has the following properties:

- i) The vertices of each edge $[cd]$ of S_1^m verify:

$$(A) \begin{cases} d \in \partial b(c, c') \text{ if } d' \leq c' \\ c \in \partial b(d, d') \text{ else} \end{cases}$$

where c' and d' are the weights associated, respectively, with the vertices c and d .

- ii) There exists in C_1^m a particular vertex c_0 so that its weight c'_0 is equal to 1.

Based on the previous properties, we define a unitary query-shape as follows:

Definition 1. *The unitary query-shape M_1^m associated with a graph S_1^m is the set:*

$$M_1^m = \cup_{c \in C_1^m} b(c, c')$$

The vertex c_0 whose weight is $c'_0 = 1$, see item ii), is called the *first center* of the query-shape M_1^m . We chose an adjacent vertex c_1 to c_0 in S_1^m . It plays the role of the *second center* of M_1^m . The weight c'_0 associated with the first center c_0 is called the *size* of the query-shape. In definition 1, the term unitary is used because the radius c'_0 of M_1^m is equal to 1.

An example of a unitary query-shape is illustrated by Fig. 1. It is a non-convex region covered with 22 discs verifying the relationship (A) item i), the dotted lines represent the graph S_1^m , the centers of the discs are the vertices of S_1^m , and there radii are the weights associated with the vertices. In this figure, many weights (radii of the discs) are equal to 1, specifically those associated with the vertices 1, 10, 14, 20 and 22. Thus, many couples of vertices can be considered as first and second centers (c_0, c_1) of the displayed query-shape, these couples are (1, 2), (10, 9), (14, 13), (20, 19) and (22, 21). The choice of one of them to play the role of the centers of the query-shape is fixed by a user according to its goal.

3.2 A query-shape of size r

A query-shape of size r is denoted M_r^m , r is a positive real number. When r is less than 1, the query-shape M_r^m shrinks the unitary one M_1^m , else it corresponds to dilation of M_1^m . M_r^m is associated with a graph S_r^m , which is computed by using the graph S_1^m . Each vertex $T(c)$ of S_r^m is obtained from a vertex c of the graph S_1^m . Formally, we define the graph S_r^m as follows:

- (i) Let us consider again the first and second centers c_0 and c_1 of the unitary query-shape M_1^m (section 3.1, after the definition 1). Let $T(c_0)$ and $T(c_1)$ be two points such that $T(c_0)T(c_1) = rc_0c_1$. $T(c_0)$ and $T(c_1)$ are the *first and second vertices of S_r^m* .
- (ii) The vertices of S_r^m are defined according to the iterative following process.

Let c be a vertex of S_1^m , and suppose that $T(c)$ is a known vertex of S_r^m , initially, $c = c_0$. Then, for each vertex d connected to c ($[cd]$ is an edge of

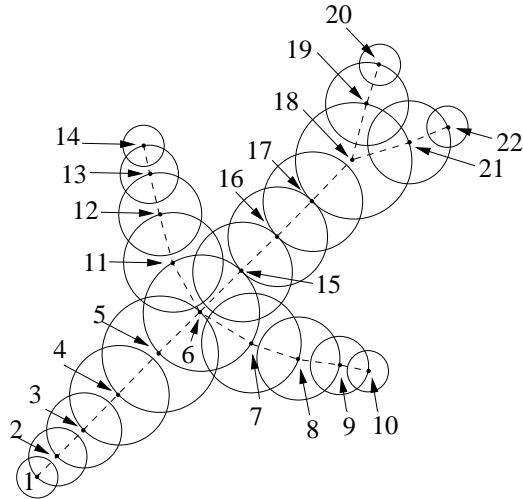


Fig. 1. A unitary query shape ($m = 22$). If the center 1 is fixed as the first center of this query shape, the point 2 is its second center.

S_1^m), we compute the vertex $T(d)$ in S_r^m using the next properties. $T(d)$ is such that the angle between the vectors $[cd]$ and $[T(c)T(d)]$ is equal to the angle between the vectors $[c_0c_1]$ and $[T(c_0)T(c_1)]$, and

$$T(c)T(d) = \begin{cases} rc' & \text{if } d' \leq c' \\ rd' & \text{else} \end{cases}$$

The radius of the disc centered at $T(d)$ is $T(d)' = rd'$.

Definition 2. The query-shape M_r^m of size r defined from the unitary query-shape M_1^m and associated with the graph S_r^m is:

$$M_r^m = \cup_{d \in C_r^m} b(d, d')$$

where C_r^m is the set of the vertices of S_r^m and d' is the radius of the disc centered at d .

The first center and the second center of M_r^m are, respectively, the vertices $T(c_0)$ and $T(c_1)$, where c_0 and c_1 are, respectively, the first and the second centers of M_1^m . Fig. 2(a), illustrates a unitary query-shape, Fig. 2(b) shows a shape of size $r = 4$ which dilates the unitary query-shape of Fig. 2(a). However, Fig. 2(c) displays a shape of size $r = 0.4$, it shrinks the unitary query-shape of Fig. 2(a). Finally, Fig. 2(d) illustrates a rotated shape.

The query-shape representation is a generalization of the disc concept, a disc of radius r can be seen as a query-shape M_r^1 of size r and consisting of one disc.

A query-shape describes a shape of a connected region, it has a size and two centers. Varying the centers, the query-shape is moved. Increasing or decreasing its size, dilates or shrinks the query-shape. The different parameters needed to compute M_r^m are:

1. **Constants: Shape parameters.** These parameters define the shape of the pattern M_1^m . They are fixed real numbers: the coordinates of the centers c and the radii c' of the discs composing M_1^m .
2. **Variables: Dilating, shrinking and moving parameters.** These parameters are the size r , the first and second centers of the shape M_r^m . Knowing the shape parameters and the two first centers of M_r^m is enough to determine the rest of centers of its discs, see the iterative process used to define S_r^m .

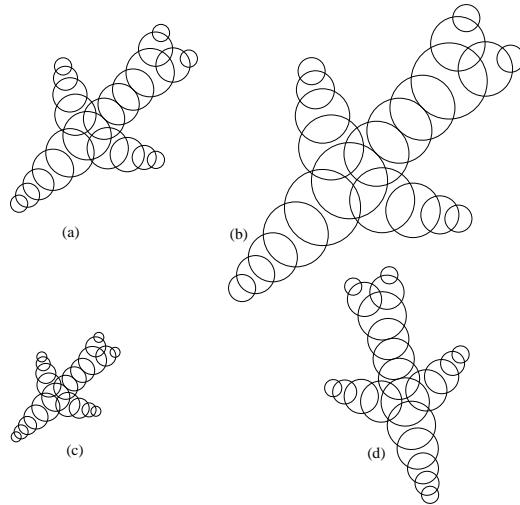


Fig. 2. (a) A unitary query-shape composed by 22 discs. (b) Dilation of the query shape presented in (a). (c) A shape obtained by shrinking the query shape (a). (d) Translation and rotation of the query shape presented in (a).

4 Retrieval patterns from a finite set of points

To detect shapes similar to a query-shape (a known region) we introduce, a generalization of the Euclidean α -shape.

4.1 (α, ϵ) -shape based on a query-shape

The goal is to retrieval, from a set of points P , the patterns similar to a query shape, independently of translation, rotation and scale changing transformations.

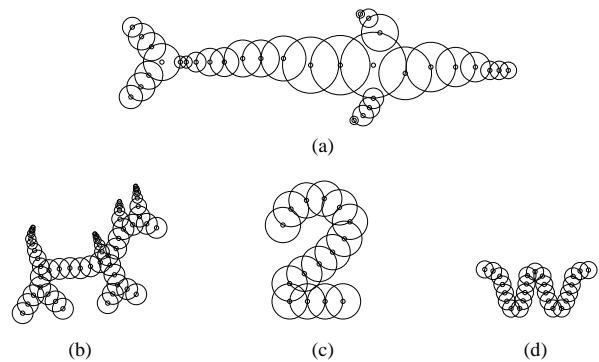


Fig. 3. Examples of four query shapes

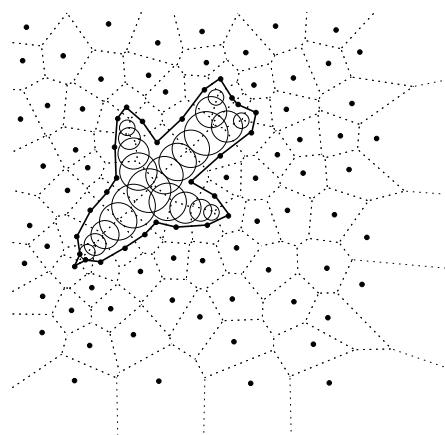


Fig. 4. A shape and its associated hull.

The proposed algorithm is based on a generalization of the Euclidean α -shape, we call the (α, ϵ) -shape based on a query shape. Before defining it, we need to introduce the following definitions. Consider a pattern M_r^m associated with the graph S_r^m , we say that M_r^m is *empty* if and only if $P \cap M_r^m = \emptyset$. We call the *hull*, $\sigma(M_r^m)$, of an empty M_r^m the open polygon of vertices in P , containing the graph S_r^m so that $\sigma(M_r^m) \cap P = \emptyset$, see Fig. 4. Let us denote K the vertices of $\sigma(M_r^m)$.

Definition 3. For two fixed positive real numbers α and ϵ , the (α, ϵ) -shape of P , based on M_1^m is the set of the hulls $\sigma(M_\alpha^m)$ associated with the empty M_α^m verifying:

- i) At least one point of P is on the boundary of M_α^m .
- ii) $\max_{v \in K} d_e(v, \partial M_\alpha^m) \leq \epsilon$.

When the query shape M_1^m is a disc and ϵ is nil then the (α, ϵ) -shape, of P , based on M_1^m becomes the Euclidean α -shape of P . The properties i) and ii) of the definition 3 as the *measure of similarity* between the query shape M_1^m and the empty patterns M_α^m . The parameter α fix the level of shrinking or dilation and the ϵ measures the similarity between the shape of an empty M_α^m and the query-shape M_1^m . Specifically, it is decided that an empty pattern M_α^m is similar to the query shape when the largest distance from the vertices of the hull $\sigma(M_\alpha^m)$ to the boundary of M_α^m is less than a tolerance ϵ .

4.2 An algorithm to retrieval patterns from a cloud of points

Knowing a query shape M_1^m , the presented algorithm extracts from a set of points P the empty patterns M_α^m that can be embedded in gaps and verifying the similarity measure defined in i) and ii) of the definition 3. This algorithm computes the (α, ϵ) -shape of P using the Voronoi diagram concept and the property of empty discs. The following observation shows how we verify that a disc is empty of the points of P .

Theorem 1. Let $b(c, r)$ be a disc centered at c and of radius r , let us p be the point so that $c \in R(p)$. $b(c, r) \cap P = \emptyset$ if and only if $p \notin b(c, r)$.

Proof. $p \notin b(c, r)$ if and only if $pc \geq r$. Thus, for all $q \in P$ we have $r \leq pc \leq qc$. Consequently, $q \notin b(c, r)$, this means $b(c, r) \cap P = \emptyset$.

To simplify the presentation of the algorithm, without loss of generality, we suppose that the graph S_1^m of the query shape M_1^m is a polygonal line. The vertices of S_1^m are denoted c_0, c_1, \dots, c_{m-1} and M_1^m is equal to $\bigcup_{k=0}^{k=m-1} b(c_k, c'_k)$, where c'_k are the radii of the discs into M_1^m . The main steps of the algorithm are briefly explained hereafter and some examples are shown in Figs. 5, 6 and 7:

Algorithm: Retrieval patterns

Input : the set of points P , the query shape $M_1^m = \cup_{k=0}^{k=m-1} b(c_k, c'_k)$, the parameters ϵ and α , the parameter t which determines the directions where the empty patterns M_α^m will be constructed. Specifically, the interval $[0..2\pi)$ is divided in t intervals $[\ell a_0..(\ell+1)a_0)$, with $ta_0 = 2\pi$. For each $\ell = 0, \dots, t-1$, each angle in the range $[\ell a_0..(\ell+1)a_0)$ is represented by the angle ℓa_0 .

Output : The empty patterns $M_\alpha^m = \cup_{k=0}^{k=m-1} b(T(c_k), \alpha c'_k)$ detected in the directions ℓa_0 , $\ell = 0, \dots, t-1$.

1. Compute the Voronoi diagram of P .

2. Compute a first empty disc of M_α^m

For each Delaunay edge $[pq]$ so that there exists an empty disc, b_α of radius α and $\partial b_\alpha \cap P = \{p, q\}$. There are at most two balls b_α . Their centers are on the Voronoi edge dual to $[pq]$. These two empty discs are the first discs of the two possible empty patterns.

3. For each computed empty disc b_α whose center is denoted $d_0 = T(c_0)$,

For each number $\ell \in \{0, 1, \dots, t-1\}$,

3.1. Compute a second empty disc of M_α^m .

Compute the point $d_1 = T(c_1)$ on the straight line of slope $\tan(\ell a_0)$, passing by d_0 , and verifying: $d_0 d_1 = \alpha c'_0 = \alpha$ if $c'_0 > c'_1$, else $d_0 d_1 = \alpha c'_1$ (see section 3.2, item (ii)).

3.2. Verify that the second disc centered at d_1 is empty

Compute the point $s \in P$ so that d_1 is into the Voronoi region $R(s)$. If s is not into the disc $b(d_1, c'_1 \alpha)$, then the second disc $b(d_1, c'_1 \alpha)$ is empty (see Theorem 1), and continue the construction of M_α^m (see next step number 4). Else, give up the construction of M_α^m in the direction ℓa_0 .

4. Compute the remaining empty discs of the pattern M_α^m

For $k \in \{2, \dots, m-1\}$,

4.1. Compute the center, $T(c_k)$, of the k th disc of M_α^m .

Compute the point $T(c_k)$ verifying: the angle between the vectors $[c_{k-1} c_k]$ and $[T(c_{k-1}) T(c_k)]$ is equal to the angle between the vectors $[c_0 c_1]$ and $[d_0 d_1]$.

4.2. Let s be the point of P such that the computed $T(c_k)$ is into the Voronoi region $R(s)$. If s is not into the disc $b(T(c_k), c'_k \alpha)$, then the disc $b(T(c_k), c'_k \alpha)$ is empty, and continue the construction of M_α^m . Else give up the construction of M_α^m in the direction ℓa_0 .

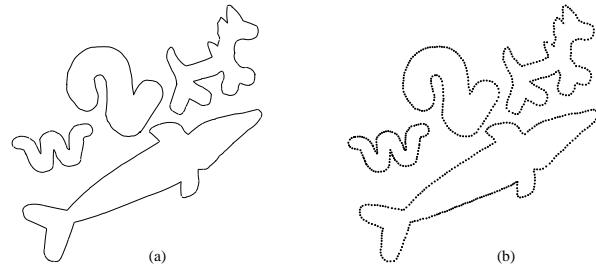


Fig. 5. (a) illustrates four patterns which are input of our algorithm. After the sampling step, we obtain a cloud of points given in (b).

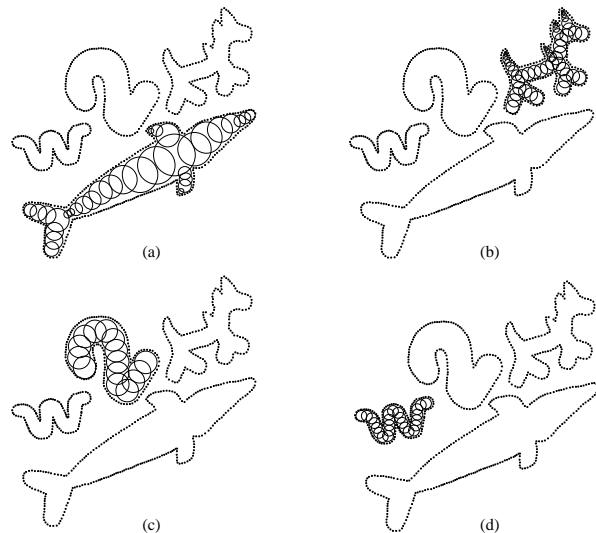


Fig. 6. This figure shows the results obtained by using sequentially the four query shapes given in Fig. 3. The used parameters are $\alpha_w = 8.2$, $\alpha_{two} = 34$, $\alpha_{dog} = 10$, $\alpha_{fish} = 31.5$ and $\epsilon = 0.6$.

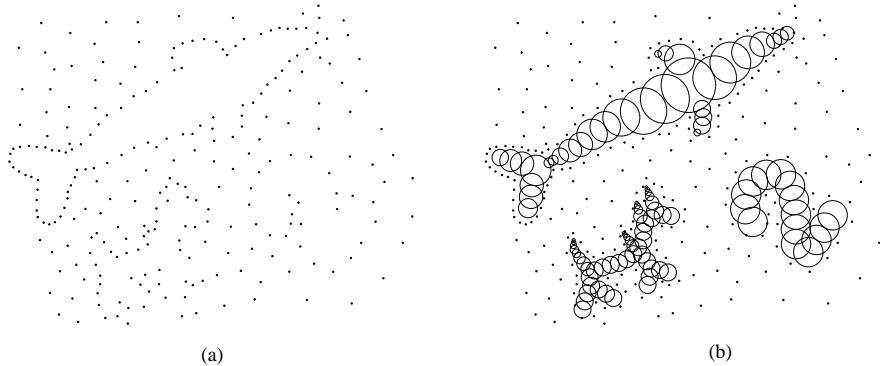


Fig. 7. A set of points and the detected patterns. (a) A set of points, (b) retrieval the patterns, similar to the query shape given in Fig. 3(a), 3(b), 3(c) and 3(d), existing in the set of points illustrated by (a), the used parameters are $\alpha_{two} = 34$, $\alpha_{dog} = 9.8$, $\alpha_{fish} = 31.5$ and $\epsilon = 0.6$.

5 Conclusion

In this paper, we have proposed a generalization of the α -shape concept. Through this new concept we have presented a geometric algorithm extracting from a cloud of points shapes that are similar to a query shape. The algorithm is accurate and it detects the overall shapes similar to the query shape. As a continuation of this work, we focus our interest on the extension of the presented algorithm to the three dimensional sets of points.

References

1. Ballard D.H.: Generalized Hough transform to detect arbitrary patterns. *IEEE Trans. Pattern Recognition and Machine Intelligence*, **13**(2) (1981) 111–122
2. Edelsbrunner H., Mücke E. P.: Three-dimensional alpha shapes. *ACM trans. on Computer Graphics* **13** (1994) 43–70
3. Khotanzad A., Hong Y. H.: Image invariant recognition by Zernike moments. *IEEE Trans. Pattern Recognition and Machine Intelligence*, **12**(5) (1990) 489–497
4. Loncaric S.: A survey of shape analysis techniques. *Pattern Recognition*, **31**(8) (1998) 983–1001
5. Okabe A., Boots B. and Sugihara K.: *Spatial tessellations : Concepts and Applications of Voronoi Diagrams*. John Wiley and Sons, England. (1992)

SHAPE PLACEMENT IN A SET OF POINTS: FINDING AND GROUPING SOLUTIONS BY USING ROTATED COPIES

MAHMOUD MELKEMI, LHASSANE IDOUMGHAR and THOMAS IWASZKO

Faculté des Sciences et Techniques, Université de Haute Alsace

4 rue des frères Lumière, 68093 Mulhouse, France

{mahmoud.melkemi, lhassane.idoumghar, thomas.iwaszko}@uha.fr

A large class of computational geometry problems deals with the placement of a given shape (usually circle, ellipse, square, rectangle, polygon) among a set of points in order to satisfy some constraint. This might be emptiness *i.e.* the shape does not contain any site, an optimization criteron or a combination of both. Here we present a similar empty shape placement problem although based on a very generic shape representation (might represent non-convex shapes and shapes with holes). We solve it for the case of rotations by using specific geometric concepts. The presented algorithm runs in $O(n \log n) + O(nm)$ time at most, where n is the number of sites and m a parameter which varies according to the shape complexity. The strength of the presented method is to find not one but all the existing solutions, and to supply a convenient mathematical representation of the solution-set which can be implemented in a very compact way. Results could be easily extended to problems involving other geometric transformations (translation, rescaling, etc.). They show an interesting link with the notion of empty disc property and hence, the well-known voronoi cells. Applications such that pattern recognition or geometric optimization are conceivable.

Keywords: shape; non-convex; disc-based; model; empty; rotation; voronoi.

1. Introduction

Since the beginning of computational geometry, back when Shamos and Hoey introduced voronoi diagrams to the world of theoretical computer science ¹, many problems involving proximity information have been discussed. One of the first known problem of this kind is the largest empty circle problem ². Interestingly enough, during the last 30 years the “proximity information” in question has increasingly become more sophisticated. To problems based on point-point distance or point-circle distance, followed problems rather dealing with distance to squares ³, rectangles ⁴, ellipses ⁵, convex polygons ⁶, etc. State of the art papers on this kind of topic go back to the 80s, *e.g.* the polygon containment problem ⁷, polygon placement under translation and rotation ^{8,9}, etc.

So finding locations satisfying certain constraints is desirable. Sometimes it is just as important to store these locations as to find them. Grouping all solutions of a proximity problem is precisely what the voronoi diagram do; it stores all the location to place a particular disc such that it contains no point. We can imagine

generalization where the constraints change, *e.g.* segments, polygons instead of the point-set. Generalized voronoi diagrams or special regions also have been proposed in order to store locations of empty copies of a different shape than a disc. We recall here the term “empty” simply means that the shape is not intersecting the environment. Again, this idea of solving placement problems while grouping all the possible solutions has been applied dozen of times^{10,11}.

Real world applications like robotics, motion planning, pattern detection, placement of obnoxious facilities etc. have been solid motivations for these studies.

To sum up, there is a clear link between all these problems. The generic statement could be: “shape placement into a static environment without intersection”. Details which vary are notably: assumptions about the object to be placed, the motions allowed for the placement and the environment the object is to be placed within. Despite of the similarity, such problems have mostly been treated in an independant manner; many specific methods exist nowadays.

Many different approaches, but as far as we know the proposed algorithms do not show capabilities for parallel computing or handling the case of a dynamic data set (shape and/or environment are changing).

In this context, a generic unified approach would have several advantages: giving more insight on the global problem by linking together existing research questions, eventually pointing out better solutions (in term of efficiency, simplicity, etc.) for existing problems, removing the need to start over a new analysis when a similar unsolved problem occur.

In this paper we introduce a novel practical approach for placement problems. It uses a special disc-based shape representation. This choice enables on one hand to use various possibly non trivial shapes (objets with holes, non-convex polygons, etc.) as they can all be approximated with several discs. On the other hand, various motions for placement *e.g.* rotation, translation, homothety are easily handled if two discs are chosen for reference. This suggests to be a good candidate for finding the unifying approach. Moreover, because of the possible decomposition of the shape-model (discs constituting the shape can be treated separately) the model is suitable for parallel computation or dynamic data sets.

In general, even in the euclidean plane, the problem space for placement problems is multidimensional. Many variables exist for determining a shape copy and its placement. It includes: position of a reference point (two dimensions), the orientation of the shape copy and its scale.

First, shape-models and their copies with respect to two reference points x_1, x_2 are defined. Then we focus on computing the locations of all empty rotated copies of a given shape. We solve the problem by providing a $O(n \log n) + O(nm)$ worst case algorithm, where n is the number of sites and m a parameter which varies according to the shape complexity. A sketch of the method is given below.

- A shape-model, a disc of center x_1 and radius r being given, we consider all the possible copies having the disc (x_1, r) for first disc. From this we

can deduce all the possible locations for x_2 (center of the second disc) ; x_2 necessarily belongs a certain circle denoted Orb_2 .

- We generalize this fact to any center x_i of the considered copies. x_i necessarily belongs to a certain circle we call the i -th orbit. After computing forbidden arcs on each orbit (arcs where the i -th disc of the copy would contain a site) we transfer all of these on Orb_2 using special transfer functions.
- The complement of: Orb_2 minus all the transferred forbidden arcs is returned. This result represents all the possible locations for x_2 such that copies with reference points x_1, x_2 are empty. In fact, this result is a set of circle-arcs whose size is at most $n \times (m - 1)$.

Solving the cases where only the scale is varying, or only the location (purely translational placement in a point set) is possible using the same shape-representation, same approach and slightly modified algorithms. These results are being formalized and written¹² in preparation of a synthetic paper showing the unified process. Hence the present paper focuses on describing with detail a general approach and gives only the very first example of its application.

The article is organized as follows: In the next section, we start by listing the notations and basic terminology used throughout the paper. A convenient shape representation based on discs is described in section 3. Thanks to its definition, we can introduce an easily computable “emptiness property” which will be of central importance in our problem. That property and the actual problem statement are given section 4. The computational resolution is then detailed in sections 5 and 6. The full algorithm proposal is made in section 7 whereas its time complexity is studied in section 8. Finally, in the last section we talk about the possible applications of our algorithm, its limitations and the perspectives.

2. Notations and Basic Terminology

Throughout the paper we use the following notations:

- $\|q - p\|$: the euclidean distance between p and q ;
- \vec{pq} : the vector from point p to point q ;
- $[pq]$: the segment of extremities p and q ;
- \widehat{pq} : a circle-arc of extremities p and q ;
- $\angle(\vec{v}_1, \vec{v}_2)$: the angle between the vectors \vec{v}_1 and \vec{v}_2 ;
- $\mathbf{b}(x, r) = \{y \in \mathbb{E}^2 \mid \|y - x\| < r\}$: the open disc of radius r centered on x ;
- $\partial\mathbf{b}(x, r) = \{y \in \mathbb{E}^2 \mid \|y - x\| = r\}$: the circle of radius r centered on x ;
- Given a set of points S , $R(p) = \{x \in \mathbb{E}^2 \mid \|x - p\| \leq \|x - q\|, \forall q \in S\}$ denotes the voronoi region of $p \in S$.

Terminology We call *weighted point* a pair constituted by a point and a real positive number, formally: $w = (p, r)$ where $p \in \mathbb{E}^2, r \in \mathbb{R}_+$. Also, for convenience we

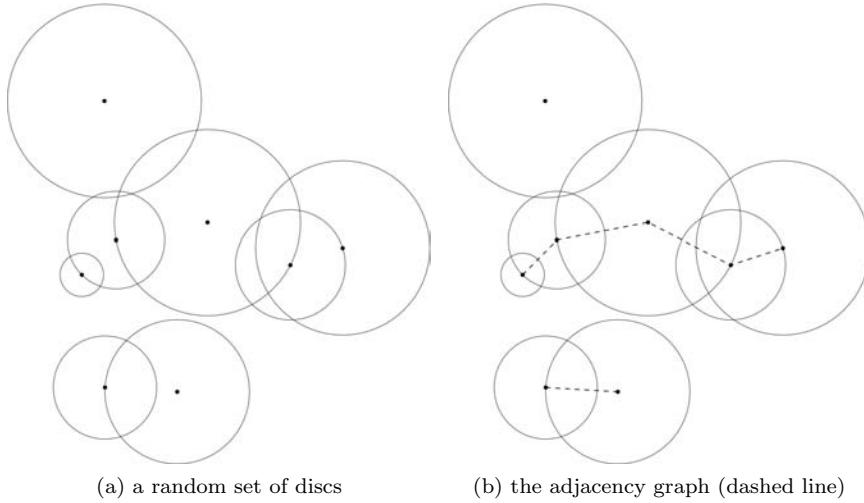


Fig. 1: Illustration of the notion of the adjacency graph. Two discs are adjacent if and only if the smallest has its center on the boundary of a largest.

shall abuse language slightly in the next section. Being given a tuple T of weighted points, *a disc of T* refers to an open disc $\mathbf{b}(c, r)$ where (c, r) is an element of T .

3. The Shape Representation

In this section, a set of definitions is proposed to formalize what the term “shape” means in our work.

The definition 1 requires two preliminary notions introduced below. Given $\mathcal{C} = ((c_1, r_1), (c_2, r_2), \dots, (c_m, r_m))$ a tuple of m weighted points:

- Two discs $(c_i, r_i), (c_j, r_j)$ of T are **adjacent discs** if and only if: the smallest has its center on the boundary of the largest, that is:

$$\begin{cases} c_j \in \partial \mathbf{b}(c_i, r_i) & \text{if } r_j \leq r_i \\ c_i \in \partial \mathbf{b}(c_j, r_j) & \text{otherwise} \end{cases}$$

- Let $V = \{c_1, \dots, c_m\}$ be the set of all the points listed in T . Let E be the set of segments $[c_i c_j]$ such that (c_i, r_i) and (c_j, r_j) from T are adjacent discs. The resulting straight-line graph (V, E) is **the adjacency graph** of T . See Fig.1 for example.

Definition 1. A tuple of m weighted points $\mathcal{C} = ((c_1, r_1), \dots, (c_m, r_m))$ is called a **shape-parameter sequence** iff it satisfies the two conditions:

- (1) The two first discs of \mathcal{C} are adjacent discs.
- (2) The adjacency graph of \mathcal{C} is connected.

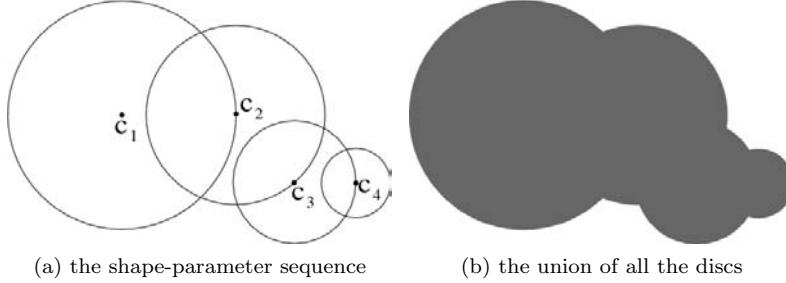


Fig. 2: An example of a four-disc shape-model. The points c_1 and c_2 are the so-called **reference points**.

Definition 2. Being given $\mathcal{C} = ((c_1, r_1), \dots, (c_m, r_m))$ a shape-parameter sequence, the **shape-model** $\mathbf{p}_m(\mathcal{C})$ is the open bounded region defined by the union of all the discs of \mathcal{C} :

$$\mathbf{p}_m(\mathcal{C}) = \bigcup_{i=1}^m \mathbf{b}(c_i, r_i)$$

Furthermore the centers of the two first discs of \mathcal{C} , namely c_1 and c_2 , are called **reference points** of the shape-model $\mathbf{p}_m(\mathcal{C})$.

Fig. 2a gives an example of a shape-parameter sequence. A shape-model is defined as the union of all the discs of such a sequence. The two conditions satisfied by the sequence ensure that either $\|c_2 - c_1\| = r_1$ or $\|c_2 - c_1\| = r_2$ is true, and that the shape-model is a connected region.

Definition 3. Given a shape-model $\mathbf{p}_m(\mathcal{C})$, its **copy with respect to two points** x_1, x_2 is the open bounded region defined by:

$$\mathbf{p}_m(\mathcal{C}, x_1, x_2) = \bigcup_{i=1}^m \mathbf{b}(c'_i, r'_i)$$

Where c'_i, r'_i are defined as follows:

- (1) $c'_1 = x_1$ and $c'_2 = x_2$
- (2) $c'_i = \lambda \mathbf{M}(c_i - c_1) + x_1$ for $1 \leq i \leq m$
- (3) $r'_i = \lambda r_i$ for $1 \leq i \leq m$
- (4) λ is the rescaling factor: $\lambda = \frac{\|x_2 - x_1\|}{\|c_2 - c_1\|}$
- (5) \mathbf{M} is the rotation matrix:

$$\mathbf{M} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \text{ where } \theta = \angle(\overrightarrow{c_1 c_2}, \overrightarrow{x_1 x_2})$$

For short, in the following we shall use the term **copy** for referring to: copy of a shape-model according to two points.

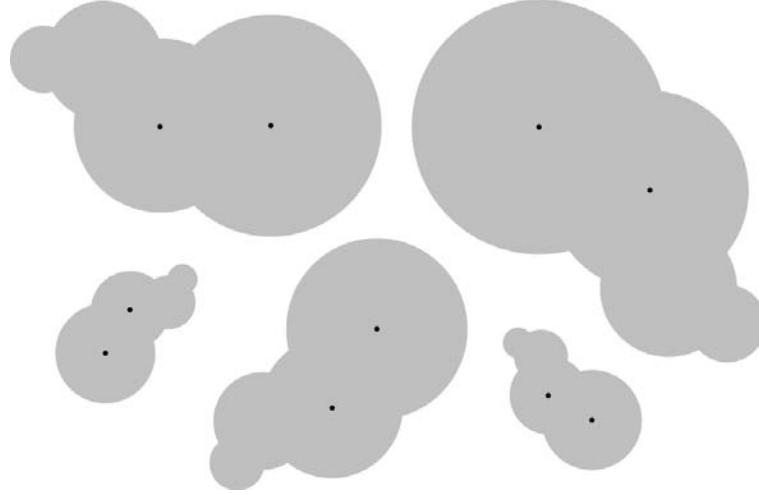


Fig. 3: Five copies of the shape-model introduced previously. For each copy, its two reference points are shown. Interestingly, it is sufficient to know the original shape-parameter sequence \mathcal{C} along with five pairs of reference points to store all these copies in memory.

Interpretation of the definition: To copy a shape-model $M_1 = \mathbf{p}_m(\mathcal{C})$ is to define a “similar” shape-model $M_2 = \mathbf{p}_m(\mathcal{C}, x_1, x_2)$ which has x_1, x_2 for reference points. Here, the term similar means that M_2 can be obtained from M_1 by combining homothety, rotation and translation.

Interpretation of the calculi: The coordinates c'_i and numbers r'_i are defined in order to perform an affine transformation whose effect is determined by x_1, x_2 . The resulting copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$ is in particular:

- A translation of $\mathbf{p}_m(\mathcal{C})$ according to the vector $\overrightarrow{c_1x_1}$, when x_1 and x_2 vary such that $\overrightarrow{x_1x_2} = \overrightarrow{c_1c_2}$.
- A homothety of $\mathbf{p}_m(\mathcal{C})$ by a factor λ , when $x_1 = c_1$ and x_2 varies such that $\overrightarrow{x_1x_2}$ and $\overrightarrow{c_1c_2}$ are colinear. λ is a real positive number. The shape-model is dilated if $\lambda > 1$ whereas it is shrunk if $\lambda < 1$.
- A rotation of $\mathbf{p}_m(\mathcal{C})$ by an angle $\angle(\overrightarrow{c_1c_2}, \overrightarrow{x_1x_2})$ and center x_1 , when $x_1 = c_1$ and x_2 varies such that $\|x_2 - x_1\| = \|c_2 - c_1\|$.

Examples of shape-model copies are given on Fig. 3.

Remark 1. As we can see in Definition 3, the shape-parameter sequence associated to $\mathbf{p}_m(\mathcal{C})$, along with a couple (x_1, x_2) is enough to compute all data c'_1, \dots, c'_m and r'_1, \dots, r'_m of a copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$. This remark is very interesting for at least two reasons:

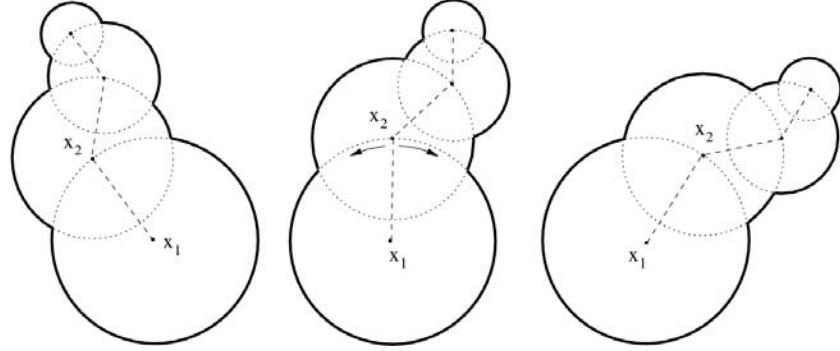


Fig. 4: We consider the copies of a given shape-model such that parameters x_1, α are fixed. Here are three examples of such copies. They share the first disc $b(x_1, r'_1)$ where r'_1 depends on α, \mathcal{C} .

- (1) *compact data storage*: For instance, a program could record a shape-model along with 50 of its copies located anywhere in the plane by just storing the unique shape-parameter sequence and 50 pairs of reference points in memory.
- (2) *representation of shape sets*: mathematical representations for grouping several shape-models can be designed thanks to this characteristic. In the following, after presenting a preliminary result, we describe how to group an infinite number of particular shape-models *via* simple circle-arcs.

4. Shape Emptiness and Problem Statement

Thanks to previous definitions, a “shape emptiness” property can now be defined mathematically. Hence, we can make explicit the problem tackled in this paper and formalize it.

Definition 4. Considering a shape-model $\mathfrak{p}_m(\mathcal{C})$ and a set of points S , $\mathfrak{p}_m(\mathcal{C})$ is said to be empty if and only if: $\mathfrak{p}_m(\mathcal{C}) \cap S = \emptyset$.

Locating free spots for a shape *i.e.* placing copies of a given shape among a set of points is a very general problem. An infinite number of cases has to be tested. In fact it is easily verified that S being a finite set, the number of possible empty shapes is infinite in the plane. The problem studied in this paper, stated below, rather assumes some parameters are fixed. We focus on rotation only. Notice that the solutions would be more of practical interest in this way, as we are not allowed to shrink the shape arbitrarily.

Problem Statement: Given a finite set of sites S , a point x_1 and a real positive number α , how to compute and record all the copies $\mathfrak{p}_m(\mathcal{C}, x_1, x_2)$ of the shape model $\mathfrak{p}_m(\mathcal{C})$ such that $\|x_2 - x_1\| = \alpha$ and $\mathfrak{p}_m(\mathcal{C}, x_1, x_2)$ is empty?

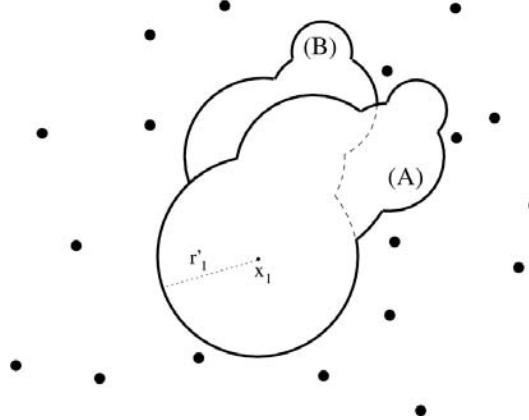


Fig. 5: The particular problem we solve is to compute and to group all the empty patterns similar to a shape model. Here all the empty patterns like (A) and (B) and similar a given shape model. We see that these shapes share the fixed first disc $\mathbf{b}(x_1, r'_1)$.

Copies with x_1 and α fixed are represented on Fig. 4. The problem is illustrated on Fig. 5. As x_1 and α are fixed, it is obvious that all the considered shape-model copies $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$ have necessarily their second reference point x_2 on the circle $\partial\mathbf{b}(x_1, \alpha)$. This fact let us use a convenient representation of the solutions, namely a set of circle-arcs and/or points, as described in the next definition.

Definition 5. Being given a point x_1 and a positive real number α , the set of points x_2 such that $\|x_2 - x_1\| = \alpha$ and the copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$ is empty, is called the **partial circle of solutions** for \mathcal{C}, x_1, α . Formally:

$$\Omega(\mathcal{C}, x_1, \alpha) = \{x_2 \in \partial\mathbf{b}(x_1, \alpha) \mid \mathbf{p}_m(\mathcal{C}, x_1, x_2) \cap S = \emptyset\}$$

In the example of Fig. 6a, the arcs of $\Omega(\mathcal{C}, x_1, \alpha)$ are the dotted ones, the bold arcs are forbidden. If the second reference point x_2 of a copy is located on the dotted arc, as in Fig. 6b, the shape copy is empty. Otherwise, x_2 is on a bold arc its associated copy is non-empty, as shown on Fig. 6c.

Remark 2. If the disc $b(x_1, \alpha)$ is not empty the problem is easily solved: there is no solution, that is $\Omega(\mathcal{C}, x_1, \alpha) = \emptyset$. From now, we only consider the non-trivial case where the first disc is empty. In practice, the first disc emptiness is checked during the initialization step of our algorithm, as explained in section 7.

5. Computation of the partial circle of solutions

How can we compute the partial circle of solutions introduced above? In this section we give three “tools” for doing it: the notion of “orbit”, the forbidden arc of a couple orbit-site and a specific affine transformation.

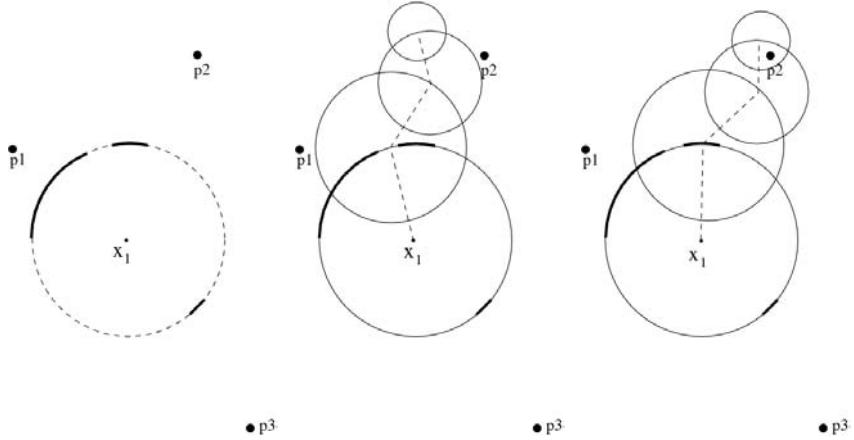


Fig. 6: Being given a shape-parameter sequence \mathcal{C} , a fixed first disc centered on x_1 of radius α , and $S = \{p_1, p_2, p_3\}$, we look for the circle-arcs which belong to the partial circle of solutions. (a) These circle-arcs are represented with dashed lines. (b) we give an exemple of an empty copy. The bold arcs are the forbidden locations for the center x_2 : they group the non empty copies of $\mathfrak{p}(\mathcal{C}, x_1, x_2)$. (c) we give an example of a non empty copy.

The idea behind orbits is the following. Firstly, considering two points c'_1, c'_i , $c'_i \in \partial\mathbf{b}(c'_1, \|c'_i - c'_1\|)$ is always true. Secondly, since we consider all the copies $\mathfrak{p}_m(\mathcal{C}, x_1, x_2)$ with x_1 and $\|x_2 - x_1\| = \alpha$ fixed, using the Def. 3 of a copy we deduce:

$$\begin{aligned} \|c'_i - c'_1\| &= \|\lambda\mathbf{M}(c_i - c_1) + x_1 - x_1\| && \text{for } 2 \leq i \leq m \\ \Leftrightarrow \|c'_i - c'_1\| &= \lambda \|c_i - c_1\| && \text{for } 2 \leq i \leq m \\ \Leftrightarrow \|c'_i - c'_1\| &= \alpha \frac{\|c_i - c_1\|}{\|c_2 - c_1\|} && \text{for } 2 \leq i \leq m \end{aligned}$$

Hence, in general the circle centered on c'_1 and passing through c'_i is known, c'_i being the i -th center of any copy described as before. This leads us to the definition of orbits.

Definition 6. The circle denoted Orb_i , centered on x_1 and passing through the i -th center c'_i of any considered copy $\mathfrak{p}_m(\mathcal{C}, x_1, x_2)$, where $\mathcal{C}, x_1, \|x_2 - x_1\| = \alpha$ are given, is called **the i -th orbit**:

$$Orb_i = \partial\mathbf{b}\left(x_1, \alpha \frac{\|c_i - c_1\|}{\|c_2 - c_1\|}\right) \quad \text{for } 2 \leq i \leq m$$

Examples of orbits are given on Fig. 7. The corresponding shape model is illustrated on Fig. 2b.

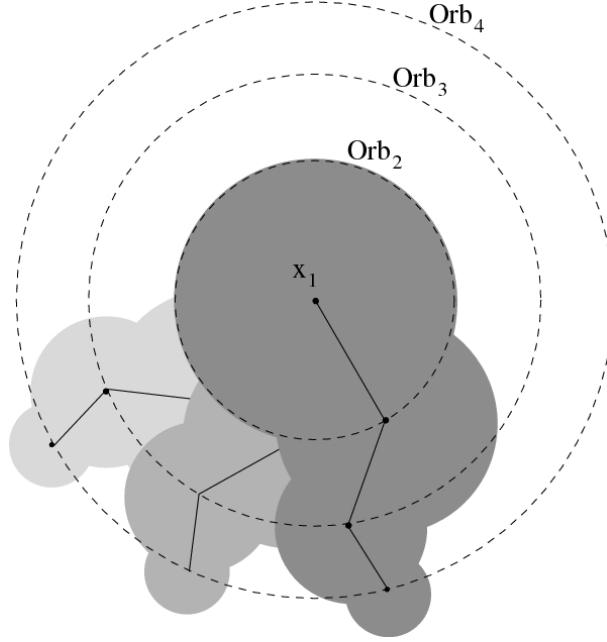


Fig. 7: Orbit are defined as being the circles $Orb_2, Orb_3, \dots, Orb_m$ passing through the points c'_2, c'_3, \dots, c'_m of all considered shape-model copies. The radius of Orb_i , $2 \leq i \leq m$ is given by the formula: $\lambda \|c_i - x_1\|$. In this example there are three orbits, represented by dashed lines. The solid poly-line starting from x_1 is the adjacency graph joining disc centers.

In the next definition, we define particular circle-arcs located on orbits. In the associated Thm. 1, it is shown in what way these circle-arcs can give information about the non-emptiness of discs that constitute shape-models.

Definition 7. Considering the i -th disc of a shape-model copy, and $p \in S$ being given, we call **forbidden arc generated by p on the i -th orbit**, the circle-arc defined by:

$$D_i(p) = Orb_i \cap \mathbf{b}(p, r'_i)$$

Moreover, it is possible to group all the forbidden arcs of a given orbit. Thus, a set of n sites $S = \{p_1, \dots, p_n\}$ being given, the set of all the forbidden arcs of the i -th orbit is defined as:

$$D_i = \bigcup_{j=1}^n D_i(p_j)$$

For instance, a forbidden arc is illustrated on Fig. 8.

Theorem 1. *Being given \mathcal{C}, x_1, α as before, we have:*

- (1) The i -th disc $\mathbf{b}(c'_i, r'_i)$ of a shape-model copy contains the site $p \in S$ if and only if c'_i belongs to the circle-arc $D_i(p)$. Formally:

$$c'_i \in D_i(p) \Leftrightarrow p \in \mathbf{b}(c'_i, r'_i)$$

- (2) The i -th disc $\mathbf{b}(c'_i, r'_i)$ of a shape-model is not empty of S if and only if c'_i belongs to the set of all the forbidden arcs of the i -th orbit. Formally:

$$c'_i \in D_i \Leftrightarrow S \cap \mathbf{b}(c'_i, r'_i) \neq \emptyset$$

In other words, D_i specifies all the locations on the i -th orbit, where the disc $\mathbf{b}(c'_i, r'_i)$ would have at least one site of S in its interior.

Proof. Let us demonstrate the two items presented in the theorem above.

- (1) Let $p \in \mathbf{b}(c'_i, r'_i)$ that is $\|c'_i - p\| < r'_i$, which is equivalent to $c'_i \in \mathbf{b}(p, r'_i)$. Considering the definition of orbits, c'_i is always on Orb_i . We conclude that $c'_i \in D_i(p) \Leftrightarrow p \in \mathbf{b}(c'_i, r'_i)$.
- (2) By definition of D_i , the statement $c'_i \in D_i$ means that there exists $p_j \in S$ such that $c'_i \in D_i(p_j)$. This fact along with the conclusion of the proof above implies $p_j \in \mathbf{b}(c'_i, r'_i)$. \square

A set of forbidden arcs D_i associated to an orbit may contain many elements. Moreover there are m sets of forbidden arcs *i.e.* as many as the number of discs of the shape-model. We see it would be quite difficult or expensive to handle them all. Could not we bring all these arcs onto a single circle: Orb_2 *i.e.* the orbit of x_2 ?

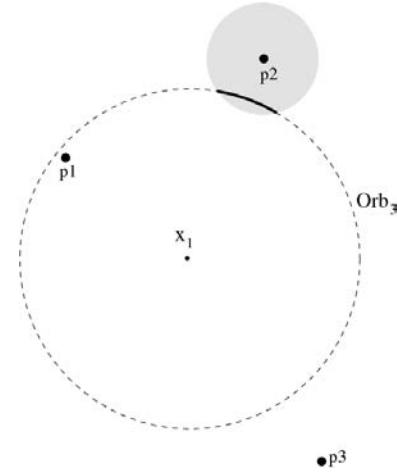


Fig. 8: The site p_2 is outside of the 3-rd disc of considered copies if and only if $c'_3 \notin \mathbf{b}(p_2, r'_3)$. Thus, the bold arc represents the forbidden arc generated by the site p_2 on the 3-rd orbit, it is denoted: $D_3(p_2)$.

Specifically, being given parameters \mathcal{C}, x_1 and a particular point z to be located on the i -th orbit, how to compute x_2 such that the copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$ verifies $z = c'_i$? The Def. 3 contains a generic formula for c'_i , namely:

$$c'_i = \lambda \mathbf{M}(c_i - c_1) + x_1 \text{ for } 1 \leq i \leq m$$

When substituting $i = 2$ it is sufficient to know the *rotation angle* *i.e.* the matrix \mathbf{M} , and the *rescaling factor* λ to compute $c'_2 = x_2$. These two informations can be computed thanks to any two points of the copy, not necessarily the two first ones. In the preceding sections, as a convention we used c'_1, c'_2 to define a copy. In general, knowing c'_1 and $c'_i = z$ for some $2 \leq i \leq m$ is sufficient to run the computation.

Thus, we define invertible affine transformations called *transfer functions*. The description and a list of properties are given below. These functions are crucial to compute the partial circle of solutions.

Definition 8. Being given x_1, α as above and a shape-parameter sequence $\mathcal{C} = ((c_1, r_1), \dots, (c_m, r_m))$, we define **transfer functions** \mathcal{T}_i for $2 \leq i \leq m$ by:

$$\begin{aligned} \mathcal{T}_i : Orb_i &\rightarrow \partial\mathbf{b}(x_1, \alpha) \\ z &\mapsto x_1 + \frac{\|z - x_1\|}{\|c_i - c_1\|} \mathbf{M}_z(c_2 - c_1) \end{aligned}$$

Where the 2×2 matrix \mathbf{M}_z , with values depending on point z , is the rotation matrix of angle $\angle(\vec{c_1c_i}, \vec{x_1z})$.

Consequences of the definition of transfer functions: These functions satisfy several properties. There is a strong link between shape-model copies and transfer functions:

- (1) $\mathcal{T}_i(c'_i)$ always maps c'_i , the i -th center of a given copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$, to the second center *i.e.* second reference point x_2 .
- (2) Considering a shape-parameter sequence \mathcal{C} and a point x_1 , it is sufficient to know either $c'_2 = x_2$ or any c'_i , $3 \leq i \leq m$ to designate a specific copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$. In the beginning as a convention, we chose $x_1, c'_2 = x_2$ as reference centers, it could be any pair x_1, c'_i as well.
- (3) The function \mathcal{T}_i is composed of translation, rotation and rescaling so it is bijective. As a result, it maps any given circle-arc $\gamma \subset Orb_i$ to a unique circle-arc $\mathcal{T}_i(\gamma) \subset Orb_2$ and since \mathcal{T}_i is a bijection, being given $\gamma \subset Orb_i$ we have:

$$c'_i \in \gamma \Leftrightarrow c'_2 \in \mathcal{T}_i(\gamma)$$

The idea of applying the transformation function on a set of points is illustrated on Fig. 9.

Remark 3. From the Thm.1 we know that there exist circle-arcs D_i on the i -th orbit such that $c'_i \in D_i(p) \Leftrightarrow p \in \mathbf{b}(c'_i, r'_i)$. The transfer functions allow us to extend this result: because of the property (P5.3), we can deduce:

$$c'_2 \in \mathcal{T}_i(D_i(p)) \Leftrightarrow p \in \mathbf{b}(c'_i, r'_i)$$

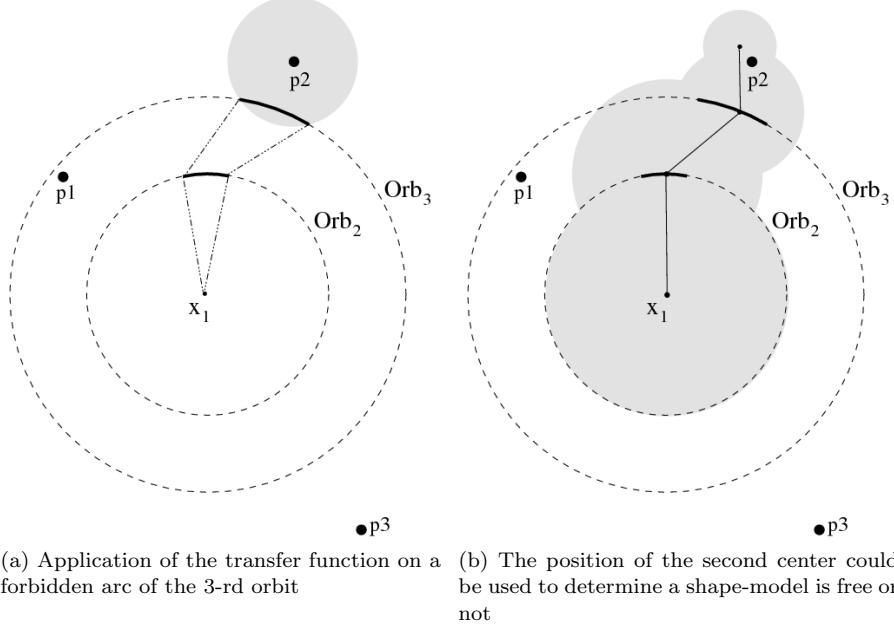


Fig. 9: Application of the transfer function T_3 to a forbidden arc of Orb_3 , in order to compute an equivalent forbidden arc on Orb_2 .

The key Theorem, used in the algorithm we will propose in section 7 is described afterwards. In order to simplify its statement and proof, being given the parameters: x_1, α and S , we assume that the first disc $\mathbf{b}(c'_1, r'_1)$ is empty of any sites of S .

This assumption is made because if the first disc is not empty it is useless running any computation. It is clear that any copy considered by the algorithm would not be empty. In practice the first disc can be checked in a prior step, this will be done in later proposed algorithms.

Theorem 2. (Main Theorem) *Given parameters \mathcal{C}, x_1, α as above, any copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$ such that $\|x_2 - x_1\| = \alpha$, is empty of S if and only if x_2 belongs to the second orbit deprived of circle-arcs which result of all transferred forbidden arcs. Formally, we have:*

$$x_2 \in \left[Orb_2 - \bigcup_{i=2}^n \mathcal{T}_i(D_i) \right] \Leftrightarrow S \cap \mathbf{p}_m(\mathcal{C}, x_1, x_2) = \emptyset$$

Proof. Let $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$ be a copy such that $\|x_2 - x_1\| = \alpha$. From Theorem 1 we know that: The i -th disc $\mathbf{b}(c'_i, r'_i)$ of a copy contains at least one site of S if and only if c'_i belongs to the set of circle-arcs D_i , where $D_i \subset Orb_i$. This can be written as:

$$c'_i \in D_i \Leftrightarrow S \cap \mathbf{b}(c'_i, r'_i) \neq \emptyset$$

The property (P5.3) of transfer functions allows us to rewrite this equivalence:

$$x_2 \in \mathcal{T}_i(D_i) \Leftrightarrow S \cap \mathbf{b}(c'_i, r'_i) \neq \emptyset$$

Since this is true for all the orbits, we can write:

$$x_2 \in \bigcup_{i=2}^m \mathcal{T}_i(D_i) \Leftrightarrow S \cap \left[\bigcup_{i=2}^m \mathbf{b}(c'_i, r'_i) \right] \neq \emptyset$$

Therefore

$$x_2 \notin \bigcup_{i=2}^m \mathcal{T}_i(D_i) \Leftrightarrow S \cap \left[\bigcup_{i=2}^m \mathbf{b}(c'_i, r'_i) \right] = \emptyset$$

Besides, we know the orbit of x_2 , so:

$$x_2 \notin \bigcup_{i=2}^m \mathcal{T}_i(D_i) \Leftrightarrow x_2 \in Orb_2 - \bigcup_{i=2}^m \mathcal{T}_i(D_i)$$

For a conclusion, because of the assumption made about the first disc and by definition of shape-model copies it is now clear:

$$x_2 \in Orb_2 - \bigcup_{i=2}^m \mathcal{T}_i(D_i) \Leftrightarrow S \cap \mathbf{p}_m(\mathcal{C}, x_1, x_2) = \emptyset$$

□

Thanks to all these concepts we know how to compute *the partial circle of solutions*. However, it might be difficult to run the computation if the number of sites S is very large. In the next section we shall demonstrate that it is possible to ignore a certain number of sites of S while still achieving correct solution, that is reduce the size of considered data set. This will result in a “smarter algorithm”, presented in section 7.

6. Data Reduction

In the problem statement, we supposed x_1 and $\alpha = \|x_2 - x_1\|$ constant. As explained in sections 3 and 5, this data is sufficient to compute the distance $\|c'_i - c'_1\|$ as well as r'_i , for any $2 \leq i \leq m$. These values are the same for all considered shape copies.

As a consequence, some sites of S may be excluded from the computation without affecting the results. In the present section, we will prove that only points of $(S \cap \mathbf{b}_{max}) \cap Ann_i$ can be included in $\mathbf{b}(c'_i, r'_i)$. The maximal disc \mathbf{b}_{max} so-called i -th annulus Ann_i will be defined.

First of all, let us recall that λ is the rescaling factor used in the definition 3 of copies, concretely: $\lambda = \frac{\|x_2 - x_1\|}{\|c_2 - c_1\|}$. Since $\alpha = \|x_2 - x_1\|$ and \mathcal{C} are known then λ can be computed without extra information. In other words, λ is fixed, it does not vary when a considered shape-model copy is rotated around $c'_1 = x_1$. We are looking for a range which *a priori* points out sites of S which would be useless in the computation.

How to do this? First of all, it is easily demonstrated that the whole shape-model $\mathfrak{p}_m(\mathcal{C})$ is included in a particular disc $\mathbf{b}(c_1, r_{max})$, where $r_{max} = \arg \max_{1 \leq i \leq m} (\|c_i - c_1\| + r_i)$. And this disc $\mathbf{b}(c_1, r_{max})$ is the smallest one. When considering copies, one has to deal with additional parameters x_1 and α but the same reasoning is working. The Def. 3 of shape-model copies allows us to say that the copy $\mathfrak{p}_m(\mathcal{C}, x_1, x_2)$ is included in the disc $\mathbf{b}(x_1, \lambda r_{max}) = \mathbf{b}\left(x_1, \frac{\alpha}{\|c_2 - c_1\|} r_{max}\right)$.

Definition 9. Being given parameters \mathcal{C} , x_1 , α , we define:

$$\mathbf{b}_{max} = \mathbf{b}\left(x_1, \frac{\alpha}{\|c_2 - c_1\|} r_{max}\right) \text{ where } r_{max} = \arg \max_{1 \leq i \leq m} (\|c_i - c_1\| + r_i)$$

Lemma 1. *Being given \mathcal{C} , x_1 and α , all the copies $\mathfrak{p}_m(\mathcal{C}, x_1, x_2)$ such that $\|x_2 - x_1\| = \alpha$ are included in the disc \mathbf{b}_{max} .*

Proof. Since $\|c'_i - c'_1\| + r'_i \leq \lambda r_{max}$ then it is clear that:

$$\mathbf{b}(x_1, \|c'_i - c'_1\| + r'_i) \subset \mathbf{b}(x_1, \lambda r_{max}) \text{ for } 1 \leq i \leq m.$$

Since

$$\mathbf{b}(c'_i, r'_i) \subset \mathbf{b}(x_1, \|c'_i - c'_1\| + r'_i),$$

then

$$\mathbf{b}(c'_i, r'_i) \subset \mathbf{b}(x_1, \lambda r_{max}) \text{ for } 1 \leq i \leq m,$$

which implies:

$$\mathfrak{p}_m(\mathcal{C}, x_1, x_2) \subset \mathbf{b}(x_1, \lambda r_{max}) \quad \square$$

Definition 10. Given problems parameters as before, we call *reduced set of sites* the set of sites of S belonging to the disc centered on $c'_1 = x_1$ in which any considered shape-model copy is included. The reduced set of sites corresponds to $S \cap \mathbf{b}_{max}$.

Prop 6.1. Let p be any site of S . If p does not belong to the reduced set of sites then any copy $\mathfrak{p}_m(\mathcal{C}, x_1, x_2)$ such that $\|x_2 - x_1\| = \alpha$ is empty of p :

$$p \notin (S \cap \mathbf{b}_{max}) \Rightarrow p \notin \mathfrak{p}_m(\mathcal{C}, x_1, x_2)$$

This proposition is a direct consequence of the Lemma 1: Any considered copy is a subset of \mathbf{b}_{max} . For instance, Fig. 10 represents the disc \mathbf{b}_{max} associated to a particular shape-model copy.

Depending on the future dataset, the previous definition and proposition help us to cut down the number of considered sites before even starting any computation. Another way of cutting down the number of sites would be to reduce S during the computation, right before performing iterative calculi. This method will be done using the “annulus” concept below.

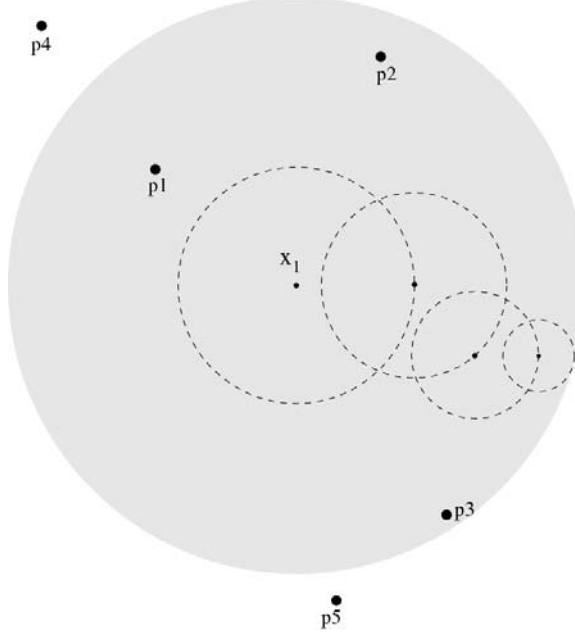


Fig. 10: The disc b_{max} is the disc in which all the considered copies are included. It can help reducing the number of considered sites. On this example, sites p_4, p_5 can be excluded of the computation of the partial circle of solutions without affecting the final result at all.

Definition 11. Being given parameters \mathcal{C}, x_1, α , we define the i -th annulus as being the area between two concentric circles centered on x_1 . Concretely:

$$Ann_i = \{x \in \mathbb{R}^2 \mid a_i < \|x_1 - x\| < A_i\}$$

where a_i, A_i are the radii of the smaller circle and the larger one, respectively. They are defined as follows:

$$\begin{aligned} a_i &= \|x_1 - c'_i\| - r'_i \\ A_i &= \|x_1 - c'_i\| + r'_i \end{aligned}$$

Fig. 11 shows an example. The annulus Ann_1, Ann_2, Ann_3 are regions of the plane associated to centers c'_2, c'_3 and c'_4 respectively.

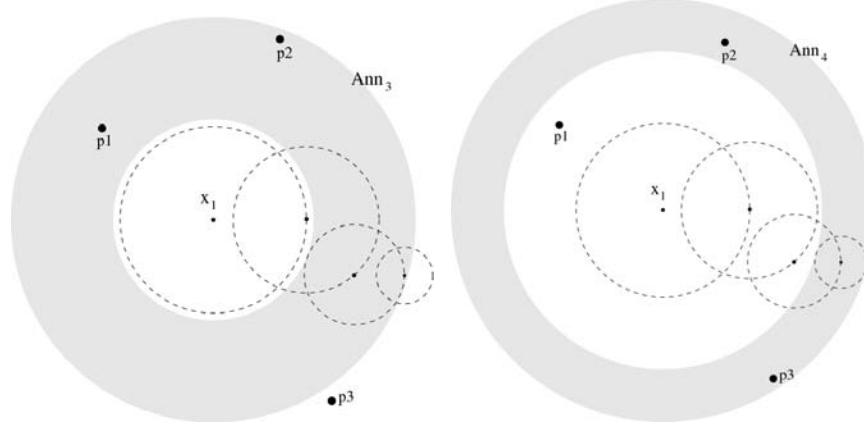


Fig. 11: Two examples of annulus Ann_3, Ann_4 (gray regions) associated to centers of considered shape-model copies. $S = \{p_1, p_2, p_3\}$. The shape model has already been shown on Fig. 1, it has three annulus in total.

Immediate properties.

- (1) Ann_i is the smallest region which includes the i -th disc $\mathbf{b}(c'_i, r'_i)$ of any considered shape-model copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$, $\|x_2 - x_1\| = \alpha$.
- (2) Of course, by definition the i -th orbit is included in the i -th annulus. That is: $Orb_i \subset Ann_i$
- (3) The i -th annulus can also be defined in the following way: $Ann_i = \bigcup_{c'_i \in Orb_i} \mathbf{b}(c'_i, r'_i)$. This is an obvious fact, as pictured on Fig. 11. Moving x_4 onto its orbit will make the last disc $b(c'_4, r'_4)$ recover the whole region Ann_4 .

These properties give us better insight on how annuli help to determine which sites of S are to be considered and which are not. Just like orbits in the preceding section, annuli give information about emptiness of discs. This is explicitized in the next proposition.

Prop 6.2. Let p be a site of S . If p does not belong to the i -th annulus Ann_i , therefore p does not belong to the i -th disc of any copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$ such that $\|x_2 - x_1\| = \alpha$. Mathematically,

$$\forall p \in S, p \notin Ann_i \Rightarrow p \notin \mathbf{b}(c'_i, r'_i)$$

The latter proposition is a direct consequence of property 6.1. As shown on Fig. 11, the sites of S which are outside the i -th annulus (the fourth one in the example) can be ignored when checking whether the i -th disc is empty of S .

In the next section, we first present a straightforward but intuitive algorithm.

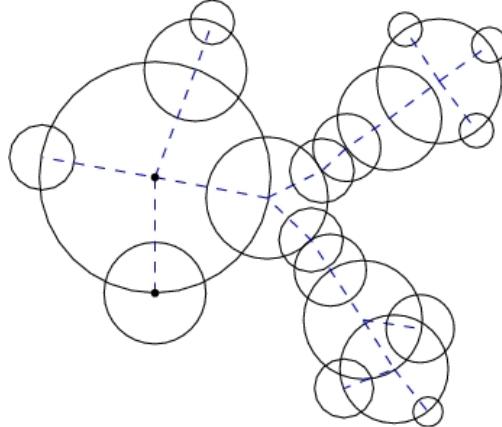


Fig. 12: Sample parameters associated to a more sophisticated shape-model. The dashed line represents the adjacency graph. It is connected according to the definition of shape parameters sequence.

Then, based on the two previous propositions 6.1 and 6.2, we will add pre-processing steps and propose a variant algorithm capable of fast computation using large input data sets.

7. Algorithm Proposal

First of all, let us recall the input/output data used by the two next algorithms presented in this section.

Input data: A finite set of n sites S , a shape-parameter sequence \mathcal{C} composed of m discs, a reference point $x_1 \in \mathbb{R}^2$ and a size $\alpha \in \mathbb{R}_+$ for the copies. α specifies the distance between x_1 and x_2 of considered copies. For instance, let us consider an arbitrary 10-point set and the shape parameters illustrated on Fig. 12. Since parameters α and x_1 are given, the first disc $b(c'_1, r'_1)$ of all considered copies is fixed.

Output data: The finite set of circle-arcs $\Omega(\mathcal{C}, x_1, \alpha)$, as defined in Def. 5, which gives all the locations where the second center $x_2, x_2 \in \partial b(x_1, \alpha)$ can be placed such that the copy $p_m(\mathcal{C}, x_1, x_2)$ remains empty of S .

The following Algorithm 1 describes how one can solve the problem stated in section 4 by using concepts presented in section 5, namely: discs, orbits and transfer functions. It is shown to ease understanding, we will however study in detail a slightly modified version, described afterwards.

As we can see in this approach, the inner *for loop* considers every site of S in order to compute forbidden arcs on a given i -th orbit. From section 6 we know that the i -th disc of any considered shape-model copy is necessarily located inside the

Algorithm 1 Straightforward Computation of the Partial Circle of Solutions

Require: a finite set S of n sites, a shape-parameter sequence $\mathbf{p}_m(\mathcal{C})$ composed of m discs, parameters x_1 and $\alpha \in \mathbb{R}_+$ for the copies.

```

if the first disc  $\mathbf{b}(x_1, \alpha)$  is not empty then
    return  $\emptyset$ 
end if
declare  $tmp$ , a circle-arc
declare  $listTFA$ , an empty list of circle-arcs
for  $i = 2$  to  $m$  do
    for  $j = 1$  to  $n$  do
        compute  $D_i(p_j) = Orb_i \cap \mathbf{b}(p_j, r'_i)$  and save it:  $tmp \leftarrow D_i(p_j)$ 
        append the transferred forbidden arc  $T_i(tmp)$ , to  $listTFA$ 
    end for
end for
merge items of  $listTFA$  that overlap, store the result in a new var.  $res$ 
return the complement  $C_{\partial \mathbf{b}(x_1, \alpha)} res$ 
```

i -th annulus Ann_i . The surface covered by Ann_i depends on the radius r_i . So if this radius is tiny then only a few sites of S are likely to be located in the disc $\mathbf{b}(c'_i, r'_i)$.

The inner *for loop* as a consequence, appears to be excessive effort in the majority of the cases.

For the purpose of improving the previous algorithm, we propose the variant Algorithm 2. It computes the solution in the precise same way, but makes special use of *pre-processing steps* that can drastically reduce the number of considered sites (compared to the size of the whole set S), depending on the dataset.

To make it clear, details on particular steps of the improved algorithm are given; we also propose to see how it works step by step on a concrete example.

Steps 1-3 These first steps check the emptiness of the first disc. If the first disc is not empty, there is no solution and there is no need to continue the calculus. The parameters x_1 and α being specified, this emptiness question boils down to test whether the distance between any site of S and the point x_1 is greater or equal to r'_1 . If it is not the case, we have: $\exists p \in S, p \in \mathbf{b}(x_1, r'_1)$ and therefore all the considered copies are not empty of S . Thus, we return $\Omega(\mathcal{C}, x_1, \alpha) = \emptyset$. Otherwise we perform the next steps.

Steps 4-5 Initialization steps. tmp is bound to store a single forbidden arc, as explained defined in Def. 7. $listTFA$ stands for *list of transferred forbidden arcs*; we start by considering that there are no forbidden arcs at all. For a given i and a given p_j , a forbidden arc associated to the i -th orbit will be computed inside the *for loops*.

Algorithm 2 Improved Computation of the Partial Circle of Solutions

Require: a finite set S of n sites, a shape-parameter sequence $\mathfrak{p}_m(\mathcal{C})$ composed of m discs, parameters x_1 and $\alpha \in \mathbb{R}_+$ for the copies.

```

1: if the first disc  $\mathbf{b}(x_1, \alpha)$  is not empty then
2:   return  $\emptyset$ 
3: end if
4: declare  $tmp$ , a circle-arc
5: declare  $listTFA$ , an empty list of circle-arcs
6: compute the reduced set:  $R \leftarrow S \cap \mathbf{b}_{max}$ 
7: sort  $R$  according to the distances:  $\|p - x_1\|$  where  $p \in R$ 
8: for  $i = 2$  to  $m$  do
9:   compute the super reduced set:  $R' \leftarrow R \cap Ann_i$ 
10:  for  $j = 1$  to  $card(R')$  do
11:    compute  $D_i(p_j) = Orb_i \cap \mathbf{b}(p_j, r'_i)$  and save it:  $tmp \leftarrow D_i(p_j)$ 
12:    append the transferred forbidden arc:  $T_i(tmp)$ , to  $listTFA$ 
13:  end for
14: end for
15: merge items of  $listTFA$  that overlap, store the result in a new var.  $res$ 
16: return the complement  $\complement_{\partial\mathbf{b}(x_1, \alpha)} res$ 

```

Step 6 A first pre-processing of S is performed to cut down the number sites considered by the whole algorithm. We know the final result is not be affected by the reduction, as explained in Def. 10 and Prop. 6.1. Again, this step is performed by simply checking the distance between x_1 and every site $p \in S$. If a distance exceeds the radius of \mathbf{b}_{max} , that is:

$$\|p - x_1\| \geq r_{max} \text{ with } r_{max} = \lambda \arg \max_{2 \leq i \leq m} (\|c_i - c_1\| + r_i)$$

then the considered site can be ignored. We recall that \mathbf{b}_{max} and r_{max} have been defined in Def. 9. We denote $R = (S \cap \mathbf{b}_{max})$ the remaining significant subset of S .

Step 7 The reduced set of sites R has to be sorted according to the distances: $\|p_j - x_1\|$ where $p_j \in R$, before entering the main loop. This will enable us to go through the computation of step 9 in an efficient way.

Step 8 The *outer for loop* goes from 2 to m , thus it considers annuli and orbits associated to all the discs of considered copies.

Step 9 The second pre-processing is made in order to reduce the number of considered sites with greater impact. It uses the annulus concept, described in section 6. It lets us know which sites of R are likely to be contained in a i -th disc of considered shape-model copies. Mathematically it means computing $R \cap Ann_i$. The step 7 already sorted R so the calculus is fast.

Step 10 This *inner for loop* is dedicated to the iterative computation of the forbidden arcs of the i -orbit, circle-arcs as described in Def. 7. It iterates over all sites $p_j \in R'$.

Step 11 Here, at each iteration we compute a single forbidden arc $D_i(p_j)$ as defined in section 5. The result is transferred right away to the orbit of x_2 , that is Orb_2 and stored in *listTFA*.

Step 15 In section 4 we mentioned that the second reference point x_2 of any copy $\mathfrak{p}_m(\mathcal{C}, x_1, x_2)$ such that $\|x_2 - x_1\| = \alpha$ is necessarily located on $\partial\mathbf{b}(x_1, \alpha)$.

In the previous steps of the algorithm we have computed all the forbidden arcs and transferred them on the orbit of x_2 . The list of all these transformed forbidden arcs is stored in a variable called *listTFA*. Now, once this computation is done, the number of items may be too large for a convenient storage in a database or for later use. Hopefully, since all the arcs belong to the same orbit Orb_2 , they can be easily merged if necessary.

Step 16 In previous steps, forbidden arcs have been computed on a certain orbit Orb_i , then transferred to Orb_2 and merged. Finally, *listTFA* stores the set of circle-arcs of Orb_2 where it is known that placing x_2 implies that the shape $\mathfrak{p}_m(\mathcal{C}, x_1, x_2)$ is *not empty*. Since we are interested in finding *empty shapes*, in the end we shall compute the complement of *listTFA*, denoted by: $\complement_{\partial\mathbf{b}(x_1, \alpha)} listTFA$.

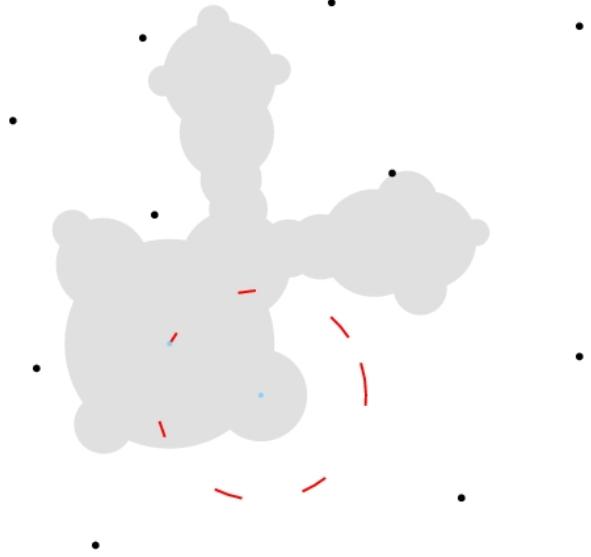
An example of results produced by this algorithm is shown on Fig. 13a and 13b. As one can see, computed circle-arcs actually encode all locations where considered copies would be empty.

Remark 4. In this section, we have proposed an algorithm which filters S in a prior pre-processing step, then iterates over discs $\mathbf{b}(c'_i, r'_i)$ in order to compute the partial circle of solutions Ω . Actually the treatment could be done in a dual way, by trying to filter the set of annulus and orbits then iterating over the sites. This algorithm would have different strengths and weaknesses. It might be subject for future work and will not be described here.

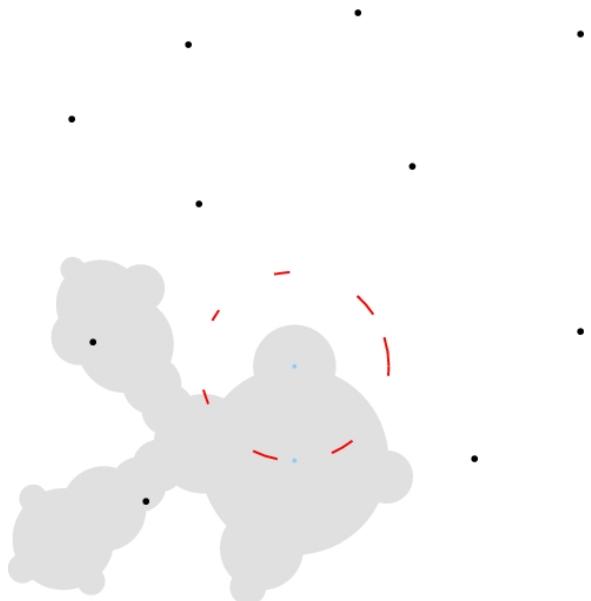
8. Algorithm Analysis

In this section we study the memory and time complexity of the Algorithm 2 presented in the previous section.

Remark 5. The last step but one of this algorithm is optional, so we won't study it in detail. When analyzing the *for loops*, one can notice forbidden arcs are simply computed and listed. Depending on the expected result (*e.g.* if one wants to use the result for later computations or store it in a database) it may be necessary run the merging step in order to try to reduce the number of stored circle-arcs in the end.



(a) x_2 does not belong to $\Omega(\mathcal{C}, x_1, \alpha)$ therefore the shape copy is empty.



(b) x_2 does not belong to $\Omega(\mathcal{C}, x_1, \alpha)$ therefore the shape copy contains at least a site of S .

Fig. 13: Example of the result computed by the proposed algorithm, the set of circle-arcs constitutes the so-called *partial circle of solutions*. The large dots represent points from S , smaller ones the two reference point of an arbitrary copy.

Theorem 3. *The number of circle-arcs in the solution set $\Omega(\mathcal{C}, x_1, \alpha)$ is at most $n \times (m - 1)$. We recall that $\Omega(\mathcal{C}, x_1, \alpha)$ records all the possible locations for x_2 such that the copy $\mathbf{p}_m(\mathcal{C}, x_1, x_2)$, $\|x_2 - x_1\| = \alpha$ is empty.*

Proof. We can construct a shape model of m discs such that the intersection I of all the annuli Ann_i , $i = 2, \dots, m$, is not empty that is: $(I = \cap_{i=2}^m Ann_i \neq \emptyset)$. Then S is chosen such that all of its n points fall into the intersection set I , see Fig. 14. Hence, for each point $p_j \in S$ we get a forbidden arc on the i -th orbit $i = 2, \dots, m$. Without merging overlapping arcs, the list of forbidden arcs is at most of length $n \times (m - 1)$. \square

Theorem 4. *Considering a set S of n sites and a shape-parameter sequence \mathcal{C} of m discs, the time complexity of the algorithm is $O(n \log n) + O(nm)$ in the worst case.*

Proof. Let us evaluate the complexity of each step of the algorithm. **Steps 1-3.** For checking the emptiness of the first disc, one has to compute the distance between x_1 and every site of S . This is done in linear time, $O(n)$. **Steps 4-5.** Constant. **Step 6.** Again, we have to check the distance between x_1 and every site of S . As seen in Def. 9, the radius of \mathbf{b}_{max} solely depends on the parameters α, \mathcal{C} and the constant r_{max} which is computed once for all, given \mathcal{C} . This result does not have to be computed at runtime, in fact it is fixed for a given shape-model. Therefore the complexity of this step is $O(n)$. **Step 7.** Any classic sorting algorithm can be used on the set R . Thus, the complexity is $O(|R| \log(|R|))$ where $|R|$ is the number of sites in R . Since the previous filtering step may not work on some degenerate datasets, in the worst case the complexity of the current step is $O(n \log n)$. **Step 9.** For computing what sites of R belong to Ann_i we have to search through R and perform test on the distance between $p_j \in R$ and x_1 in order to determine whether it is in the range $[a_i, A_i]$ or not. Since R is already sorted, this step is done in

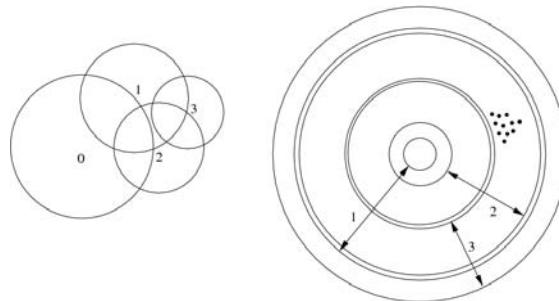


Fig. 14: Worst-case scenario; neither the maximal disc \mathbf{b}_{max} nor annuli A_i can help to reduce the set of forbidden circle-arcs to be computed.

linear time with respect to the size of R . This step is performed once for each annulus (or such that $1 \leq i \leq m$). Consequently the total complexity of this step is $O(mn)$. Let us recall that we denote R' the result of this second data reduction step. **Step 11.** Operation done in constant time, for each site of R' and each i -th orbit. The complexity of this step is also $O(mn)$. **Step 12.** Identical to the previous step. **Step 15.** This step is optional therefore it is not taken into account in our complexity analysis. It can be required depending on the expected result and the use of results made further. Please note that this step could be realized quite efficiently by sorting the min and max boundaries of all intervals. **Step 16.** For computing the complement, one has to look through the whole list res (or $listTFA$ if the merging step was not realized) and inverting the limit points. Anyway, since the size of res is $m \times n$ in the worst case, as demonstrated in Thm. 3, the complexity of this last step is $O(mn)$ at most. \square

9. Conclusion

This paper focused on aspects of the shape placement problem in a set of points. Specifically, we considered the search for shape copies that are empty (do not contain any site) using the rotation. We started by proposing a general and convenient disc-based shape representation, and defining the objective formally. Then we introduced novel concepts that helped us not only to find a solution, but to compute all existing solutions, to represent them in a simple structure (a set of circle-arcs) and to regroup them (simply with union set-operator). Two algorithms have been proposed, the latter being optimized for working on large datasets as it adds extra pre-processing data reduction steps to calculii. These results are viewed by authors as a promising lead for harder problems of shape placement *i.e.* searching empty shape copies using different/more geometric operations. The simplicity of the approach along with the fact it uses exclusively geometric objects and principles are an argument in favor of the assertion the results can be extended and generalized. Since shapes are represented by a list of discs, and that results of calculus can be encoded and stored somehow mathematically, an interesting topic is considering the possibility to define a set of voronoi-like regions satisfying emptiness properties. This related work is under investigation and has already lead to substantial results. Although this work has been presented from a theoretical point of view, applications to the fields of pattern recognition and optimization are subject for future work.

References

1. Michael Ian Shamos, Dan Hoey, "Closest-point problems". 16th Annual Symposium on Foundations of Computer Science (FOCS 1975), 1975, pp.151-162
2. Toussaint G. T., "Computing largest empty circles with location constraints". Int. J. Comput. Info. Sci. 12(5), (1983), pp. 347-358.
3. Lee D. T., and Wong C. K., "Voronoi Diagrams in L_1 (L_∞) Metrics with 2-Dimensional Storage Applications", SIAM J. Computing, volume 9,(1980), p.200-211.

4. B. Chazelle, R.L. Drysdale III and D.T. Lee, “Computing the Largest Empty Rectangle”. SIAM J. Comput., 15(1), (1986), pp. 300-315.
5. R. Dwyer and W. Eddy, “Maximal empty ellipsoids”. Int. J. of Comput. Geometry and Applications, 6, (1996), pp. 169-186.
6. Chew L. P. and Drysdale R. L., “Voronoi diagrams based on convex distance functions”. 1st ACM Sympos. Comput. Geom., (1985), pp. 235-244
7. B. Chazelle, “The Polygon Containment Problem”. Advances in Computing Research 1 (F.P. Preparata, ed.), JAI Press (1983), pp. 1-33.
8. F. Aynaim and J-D. Boissonnat, “Polygon Placement under Translation and Rotation”. In Proc. 5th Sympos. Theoret. Aspects Comput. Sci., volume 294 of Lecture Notes in Computer Science, (Springer-Verlag, 1988), pp. 322-333.
9. L. Paul Chew, Klara Kedem, “Placing the Largest Similar Copy of a Convex Polygon Among Polygonal Obstacles”. Proceedings of the Fifth Annual Symposium on Comput. Geometry (SoCG), (1989), pp. 167-173.
10. D. Leven and M. Sharir, “Planning a purely translational motion for a convex object in twodimensional space using generalized Voronoi diagrams”. Discrete Comput. Geom., 2, (1987), pp. 9-31.
11. D. Halperin, M. Sharir, “A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment”. J. Discrete and Comput. Geometry, 16(2), (1996), pp. 121-134.
12. T. Iwaszko, M. Melkemi, L. Idoumghar, “A Framework for placing an arbitrary shape among a set of points”. Technical Report, LMIA-MAGE. Universite Haute Alsace. July 2010.

Annexe : Quelques unes de mes publications
