

Proceedings of the International Conference on Swarm Intelligence Based Optimization (ICSIBO'2014)

13, 14 May 2014
Mulhouse, France



Sponsored By:



Foreword

These proceedings include the papers presented at the International Conference on Swarm Intelligence Based Optimization, ICSIBO'2014, held in Mulhouse (France). ICSIBO'2014 is a continuation of conferences OEP'2003 (Paris), OEP'2007 (Paris) and ICSI'2011 (Cergy-Pontoise).

The aim of this conference is to highlight the theoretical progress of swarm intelligence metaheuristics and their applications.

Swarm Intelligence is a computational intelligence technique involving the study of collective behavior in decentralized systems. Such systems are made up of a population of simple individuals interacting locally with one another and with their environment. Although there is generally no centralized control on the behavior of individuals, local interactions among individuals often cause a global pattern to emerge. Examples of such systems can be found in nature, including ant colonies, animal herding, bacteria foraging, bee swarms, and many more.

Authors had been invited to present original work relevant to Swarm Intelligence, including, but not limited to: Theoretical advances of swarm intelligence metaheuristics ; Combinatorial, discrete, binary, constrained, multi-objective, multi-modal, dynamic, noisy, and large-scale optimization ; Artificial immune systems, particle swarms, ant colony, bacterial foraging, artificial bees, fireflies algorithm ; Hybridization of algorithms ; Parallel/distributed computing, machine learning, data mining, data clustering, decision making and multi-agent systems based on swarm intelligence principles ; Adaptation and applications of swarm intelligence principles to real world problems in various domains, including medicine, biology, chemistry, finance, insurance, economics, social sciences, transportation, tourism, education, defense, telecommunications, energy, management, information retrieval, software engineering, fraud detection, environment, remote-sensing, robots.

Each submitted paper has been reviewed by three members of the International Program Committee. Among the 48 submissions received, 18 papers and 12 abstracts have been selected for oral presentation. A selection of the best papers presented at the conference and further revised will be published as a volume of Springer's LNCS series. The highest quality papers will be proposed for quick publication in Springer's Swarm Intelligence.

We would like to express our sincere gratitude to our invited speakers: Nicolas Monmarché and Maurice Clerc.

The success of the conference resulted from the input of many people to whom we would like to express our appreciation: the members of Program Committee and the secondary reviewers for their careful reviews that ensure the quality of the selected papers and of the conference.

We take this opportunity to thank the different partners whose financial and material support contributed to the organization of the conference: Université de Haute Alsace, Faculté des Sciences et Techniques, ROADEF, GDR-MACS. Last but not least, we thank all the authors who have submitted their research papers to the conference, and the authors of accepted papers who attend the conference to present their work. Thank you all.

P. Siarry, L. Idoumghar and J. Lepagnot
Organizing Committee Chairs

Symposium organization

Organizing Committee Chairs	P. Siarry, L. Idoumghar and J. Lepagnot
Program Chair	M. Clerc
Website / Proceedings / Administration	MAGE Team, LMIA Laboratory

Scientific committee

Ajith Abraham	Norwegian University of Science and Technology, Norway
Amitava Chatterjee	University of Jadavpur, Kolkata, India
Raymond Chiong	University of Newcastle, Australia
Maurice Clerc	Independent Consultant, France
Carlos A. Coello Coello	CINVESTAV-IPN, Depto. de Computacion México
Pierre Collet	Université de Strasbourg, France
Jean-Charles Créput	University of Technologie Belfort-Montbéliard, France
Luca Di Gaspero	University of Udine, Italy
Frédéric Guinand	Université du Havre, France
Rachid Ellaia	Mohammadia School of Engineering, Morocco
Said Hanafi	Université de valenciennes, France
Jin-Kao Hao	Université d'Angers, France
Vincent Hilaire	Université de Technologie de Belfort-Montbéliard, France
Lhassane Idoumghar	Université de Haute-Alsace, France
Jim Kennedy	Bureau of Labor Statistics, Washington, USA
Peter Korosec	University of Primorska, Koper, Slovenia
Abderrafiaâ Koukam	University of Technologie Belfort-Montbéliard, France
Mohamed G. Omran	Gulf University for Science and Technology, kuwait
Fred Glover	OptTek Systems, Inc, Boulder, Colorado, USA
Bijaya Ketan Panigrahi	Indian Institute of Technology, India
Fabrice Lauri	Université de Technologie de Belfort-Montbéliard, France
Stephane Le Menec	RGNC at EADS / MBDA, France
Julien Lepagnot	Université de Haute-Alsace, France
Nicolas Monmarché	Université François Rabelais Tours, France
René Natowicz	ESIEE, France
Abraham P. Punnen	Simon Fraser University, Canada
Monia Rekik	Université Laval, Canada
Said Salhi	CLHO Kent Business School, UK
Andrea Schaerf	DIEGM, University of Udine, Italy
René Schott	University of Lorraine, France
Patrick Siarry	Université de Paris-Est Créteil, France
Ponnuthurai N. Suganthan	Science and Technology University, Singapore
Eric Taillard	University of Applied Sciences of Western Switzerland
El-Ghazali Talbi	Polytech'Lille, Université de Lille 1, France
Antonios Tsourdos	Defence Academy of the United Kingdom, UK
Mohamed Wakrim	University of Ibou Zohr, Agadir, Morocco
Rolf Wanka	University of Erlangen-Nuremberg, Germany

ICSIBO'2014 Participants

Abdelkafi Omar	Université de Haute Alsace, France
Adam Bruno	Université de Haute Alsace, France
Aiboud Fazia	Université Blaise Pascal, France
Ait-Adda Samia	Ecole nationale supérieure d'informatique, Algeria
Aupetit Sébastien	Université de Tours, France
Bassimir Bernd	Friedrich-Alexander-University Erlangen-Nuremberg, Germany
Bedoui Akram	Université de Lorraine, France
Benaichouche Ahmed-Nasreddine	Université de Paris-Est, France
Boudjemaa Redouane	University Mhamed Bougara of Boumerdes, Algeria
Chamoret Dominique	Université de Technologie de Belfort-Montbéliard, France
Clerc Maurice	Independent Consultant, France
Deroussi Laurent	Université Blaise Pascal, France
El-Mourabit Yousef	Ecole Nationale des Sciences Appliquées d'Agadir, Agadir, Morocco
Hmaidi Kawther	Institut supérieur de gestion de Tunis, Tunisia
Idoumghar Lhassane	Université de Haute Alsace, France
Jandaud Pierre-Olivier	Université de Valenciennes, France
Kao Yucheng	Tatung University, Taiwan
Koukam Abderrafiaâ	Université de Technologie de Belfort-Montbéliard, France
Lahsinat Yasmine	Université des Sciences et de la Technologie Houari Boumediene, Algeria
Lauri Fabrice	Université de Technologie de Belfort-Montbéliard, France
Le Van-Minh	Vietnam National University, Hanoi, Vietnam
Leboucher Cédric	Missile systems, defence systems (MBDA), France
Lemoine David	Ecole des Mines de Nantes, France
Lepagnot Julien	Université de Haute Alsace, France
Lersteau Charly	Université de Bretagne Sud, France
Maillot Yvan	Université de Haute Alsace, France
Melkemi Mahmoud	Université de Haute Alsace, France
Monmarché Nicolas	Université de Tours, France
Pierre-Alain Muller	Université de Haute Alsace, France
Omran Mahamed	Gulf University for Science & Technology, Kuwait
Parrend Pierre	ECAM Strasbourg-Europe, France
Poormirzaee Rashed	Sahand University of Technology, Iran
Riad Menasri	Université de Paris-Est, France
Salmon Sebastien	Optimization Command & Control Systems, France
Schmitt Dominique	Université de Haute Alsace, France
Schott René	Université de Lorraine, France
Siarry Patrick	Université de Paris-Est, France
Sioud Aymen	Université du Québec à Chicoutimi, Canada
Wang Hongjian	Université de Technologie de Belfort-Montbéliard, France
Zahir Sahli	Université de Valenciennes, France

ICSIBO'2014 Accepted papers

1. Pierre-Olivier Jandaud, Johakim Corroyer, Souad Harmand, Said Hanafi, Abdelghani Bekrar and Mohammed Bellalij. *Aero-Thermal Optimization of a Heat Sink using Particle Swarm Optimization*
2. Bernd Bassimir, Manuel Schmitt and Rolf Wanka. *How Much Forcing is Necessary to Let the Results of Particle Swarms Converge?*
3. Riad Menasri, Hamouche Oulhadj, Boubaker Daachi, Amir Nakib and Patrick Siarry. *Smooth trajectory planning for robot using particle swarm optimization*
4. Rashed Poormirzaee, Rasoul Hamidzadeh Moghadam, Ahmad Zarean. *PSO: A Powerful and Fast Intelligence Optimization method In Processing of Passive Geophysical Data*
5. Redouane Boudjemaa. *Accelerated Particle Swarm Optimisation: Weather Network Architecture Case Study*
6. Dominique Chamoret, Sébastien Salmon, Noëlie Di Cesare and Xu Yingjie. *BSG-Starcraft Radius improvements of Particle Swarm Optimization Algorithm: an application to Ceramic Matrix Composites*
7. Yucheng Kao, Ming-Hsien Chen and Kai-Ming Hsieh. *Combining PSO and FCM for Dynamic Fuzzy Clustering Problems*
8. Akram Bedoui. *Hybrid ACO-SA algorithm for the Multi-objective Frequency Assignment Problem in Broadcasting*
9. Zahir Sahli, Abdelghani Bekrar and Damien Trentesaux. *Hybrid PSO algorithm for the solving of the Optimal Reactive Power Problem*
10. Cédric Leboucher, Patrick Siarry, Stéphane Le Ménec, Hyo-Sang Shin, Antonios Tsourdos and Rachid Chelouah. *An enhanced Particle Swarm Optimisation algorithm combined with Neural Networks to decrease computational time*
11. Laurent Deroussi. *An hybrid PSO applied to Flexible Manufacturing Systems*
12. Nicolas Monmarché, Sébastien Aupetit, Pierre Gaucher and Mohamed Slimane. *A study of task division of ants for online tuning of metaheuristic parameters*
13. Alina Mereuta, Sébastien Aupetit, Nicolas Monmarché and Mohamed Slimane. *Using bio-inspired algorithm to compensate web page color contrast for dichromat users*
14. Sébastien Aupetit, Nicolas Monmarché and Mohamed Slimane. *Comparison of two swarm intelligence optimization algorithms on the textual color problem for web accessibility*
15. Ahmed Nasreddine Benaichouche, Hamouche Oulhadj and Patrick Siarry. *Image post-segmentation correction using ant colony optimization*
16. Hongjian Wang and Jean-Charles Créput. *Parallel and Distributed Implementation Models for Bio-inspired Optimization Algorithms*

17. Omar Abdelkafi, Julien Lepagnot and Lhassane Idoumghar. *Multi-level parallelization for hybrid ACO*
18. David Lemoine. *Particle Swarm Optimization for the Multi-Level Lot Sizing Problem*
19. Pierre Parrend, Cecilia Zanni-Merk, Pierre Collet and Pierre Masai. *Swarm projects: beyond the metaphor*
20. Jiawei Zhu, Fabrice Lauri, Abderrafaa Koukam and Vincent Hilaire. *Fuzzy Logic Control Optimized by Artificial Immune System for Building Thermal Condition*
21. Kawther Hmaidi and Jalel Akaichi. *Floods Trajectories Modeling and Dynamic Relief Planning: A Bees' Foraging Approach*
22. Yasmine Lahsinat and Dalila Boughaci. *Improving Biogeography Based Optimization by Using Stochastic Local Search and Solving MI-FAP Problem in GSM Networks*
23. Fazia Aiboud, Nathalie Grangeon and Sylvie Norre. *Various heuristics with cellular automaton to generate 2D shapes*
24. Van-Minh Le, Yann Chevalere, Tuong Vinh Ho and Jean-Daniel Zucker. *A general approach to solve decomposable optimization problems in multiagent simulations settings: application to tsunami evacuation*
25. Yousef El Mourabit, Ahmed Tomanari, Anouar Bouirden, Hicham Zougagh and Rachid Latif. *Intrusion Detection System In Mobile Ad Hoc Network Based On Mobile Agent*
26. Ait Adda Samia and Balla Amar. *The use of ontology in semantic analysis of the published learner's messages for adaptability*
27. Charly Lersteau, Marc Sevaux and André Rossi. *Multiple Mobile Target Tracking in Wireless Sensor Networks*
28. Fabrice Lauri and Abderrafaa Koukam. *Robustness Analysis of Multi-Agent Patrolling Strategies using Reinforcement Learning*
29. Aymen Sioud, Caroline Gagné and Marc Gravel. *A new ACO for solving a hybrid flexible flowshop with sequence-dependent setup times*
30. Mahamed Omran, Maurice Clerc, Ayed Salman and Salah Alsharhan. *A Fuzzy-Controlled Comprehensive Learning Particle Swarm Optimizer*

ICSIBO'2014 Scientific program

Tuesday, 13 May 2014

08:00 *Welcome*

08:40 *Plenary session 1*
Nicolas Monmarché

Chair: Maurice Clerc
*Artificial ants: collective intelligence in
computing environments*

09:50 *Coffee Break*

Session 1: Particle Swarm Optimization Chair: Patrick Siarry

10:20 Pierre-Olivier Jandaud

Aero-Thermal Optimization of a Heat Sink using Particle Swarm Optimization

10:40 Bernd Bassimir

How Much Forcing is Necessary to Let the Results of Particle Swarms Converge?

11:00 Riad Menasri

Smooth trajectory planning for robot using particle swarm optimization

11:20 Rashed Poormirzaee

PSO: A Powerful and Fast Intelligence Optimization method In Processing of Passive Geophysical Data

11:40 Dominique Chamoret

BSG-Starcraft Radius improvements of Particle Swarm Optimization Algorithm: an application to Ceramic Matrix Composites

12:00 Pierre Parrend

Swarm projects: beyond the metaphor

12:20 *Lunch Break*

Session 2: Hybrid metaheuristics

Chair: Lhassane Idoumghar

14:00 Yucheng Kao

Combining PSO and FCM for Dynamic Fuzzy Clustering Problems

14:20 Akram Bedoui

Hybrid ACO-SA algorithm for the Multi-objective Frequency Assignment Problem in Broadcasting

14:40 Zahir Sahli

Hybrid PSO algorithm for the solving of the Optimal Reactive Power Problem

15:00 Cédric Leboucher

An enhanced Particle Swarm Optimisation algorithm combined with Neural Networks to decrease computational time

15:20 Laurent Deroussi

An hybrid PSO applied to Flexible Manufacturing Systems

15:40 *Coffee Break*

Session 3: Ant colony optimization

Chair: Julien Lepagnot

16:10 Nicolas Monmarché, Sébastien Aupetit

A study of task division of ants for online tuning of metaheuristic parameters

16:30 Sébastien Aupetit, Nicolas Monmarché

Using bio-inspired algorithm to compensate web page color contrast for dichromat users

16:50 Sébastien Aupetit, Nicolas Monmarché

Comparison of two swarm intelligence optimization algorithms on the textual color problem for web accessibility

17:10 Ahmed Nasreddine Benaichouche

Image post-segmentation correction using ant colony optimization

17:40 *Visit & Dinner in Ribeauvillé*

Wednesday, 14 May 2014

08:40 *Plenary session 2*
Maurice Clerc

Chair: Nicolas Monmarché
List-Based Optimisers

10:15 *Coffee Break*

Session 4: Distributed algorithms

Chair: René Schott

10:40 Hongjian Wang
Parallel and Distributed Implementation Models for Bio-inspired Optimization Algorithms
11:00 Omar Abdelkafi
Multi-level parallelization for hybrid ACO
11:20 David Lemoine
Particle Swarm Optimization for the Multi-Level Lot Sizing Problem
11:40 Redouane Boudjemaa
Accelerated Particle Swarm Optimisation: Weather Network Architecture Case Study

12:00 *Lunch Break*

Session 5: Artificial systems

Chair: Laurent Deroussi

13:45 Fabrice Lauri
Fuzzy Logic Control Optimized by Artificial Immune System for Building Thermal Condition
14:05 Kawther Hmaidi
Floods Trajectories Modeling and Dynamic Relief Planning: A Bees' Foraging Approach
14:25 Yasmine Lahsinat
Improving Biogeography Based Optimization by Using Stochastic Local Search and Solving MI-FAP Problem in GSM Networks
14:45 Fazia Aiboud
Various heuristics with cellular automaton to generate 2D shapes

15:05 *Coffee Break*

Session 6: Multi-agent systems

Chair: Abderrafiaâ Koukam

15:35 Van-Minh Le
A general approach to solve decomposable optimization problems in multiagent simulations settings: application to tsunami evacuation
15:55 Yousef El Mourabit
Intrusion Detection System In Mobile Ad Hoc Network Based On Mobile Agent
16:15 Ait Adda Samia
The use of ontology in semantic analysis of the published learner's messages for adaptability
16:35 Charly Lersteau
Multiple Mobile Target Tracking in Wireless Sensor Networks
16:55 Fabrice Lauri
Robustness Analysis of Multi-Agent Patrolling Strategies using Reinforcement Learning

17:15 *Conference end*

Accepted
papers and
abstracts

Aero-Thermal Optimization of a Heat Sink using Particle Swarm Optimization

P.O. JANDAUD^{a,b,*}, J. CORROYER^{a,b}, S. HARMAND^{a,b}, S. HANAFT^{a,c}, A. BEKRAR^{a,b}, M. BELLALIJ^{a,d}

^a Université de Lille Nord-de-France, F-59000 Lille, France

^b TEMPO EA 4542, Université de Valenciennes, F-59313 Valenciennes, France

^c LAMIH UMR CNRS 8201, Université de Valenciennes, F-59313 Valenciennes, France

^d LAMAV EA 4015, Université de Valenciennes, F-59313 Valenciennes, France

* Corresponding author: pojandaud@gmail.com

Extended Abstract

Heat sinks are devices widely used for cooling electronic components. Their principle is to increase the contact area between air and the heating surface with fins or pins. On car alternators, a heat sink is placed on the back of the machine to cool the rectifier bridge and it also allows air to enter the machine to cool the windings of the stator and the rotor.

As a result, a good heat sink should dissipate the heat produced by the electronic components but also should not affect too much the air flow. Its design is a multi-objective optimization problem with two objectives to minimize: the head loss of the system and its thermal resistance. Head loss (in Pa) is the difference of total pressure between the two ends of the system and represents the losses of mechanical energy mostly due to friction. Thermal Resistance (in K/W) is the ratio of the temperature rising to total heat flux and measures directly the thermal performances of the heat sink.

The optimization is performed using Particle Swarm Optimization with an analytical aero-thermal model of the heat sink. Five parameters are selected for the optimization (Fig. 1): the fins thickness (t), the fins spacing (a), the fins height (b), the number of fins (n) and the base thickness (h). Three different materials are considered: copper, aluminum and steel. Three cases are also studied. In the first one, the objective function is the thermal resistance and the head loss is constrained. In the second one, the objective function is now the head loss and the thermal resistance is constrained. Finally, in the third one, an aggregated objective function (AOF) is used.

By using different values of constraints (first two cases) and weights (for the AOF), a Pareto front of the optimal geometries of the heat sink is generated (Fig. 2). The optimization results also show the influence of certain parameters compared to others and provide a good starting point to design, in the future, more complex heat sinks.

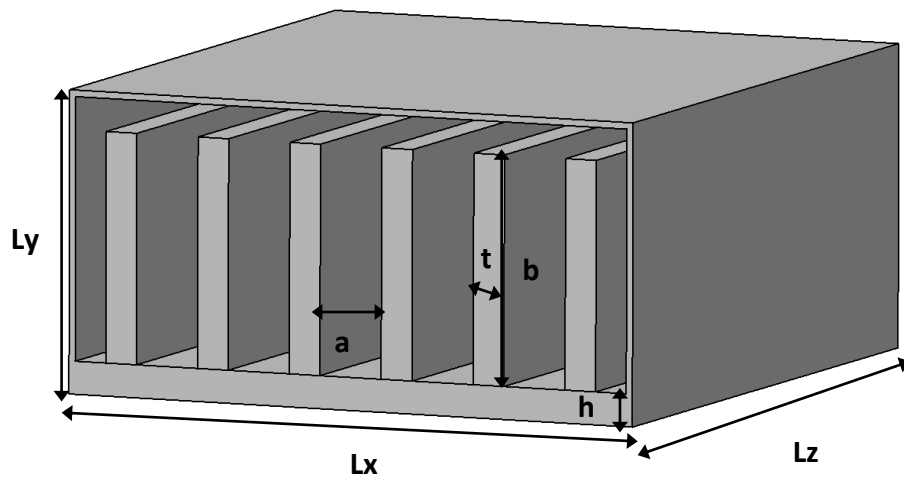


Figure 1. Geometry of the Heat Sink

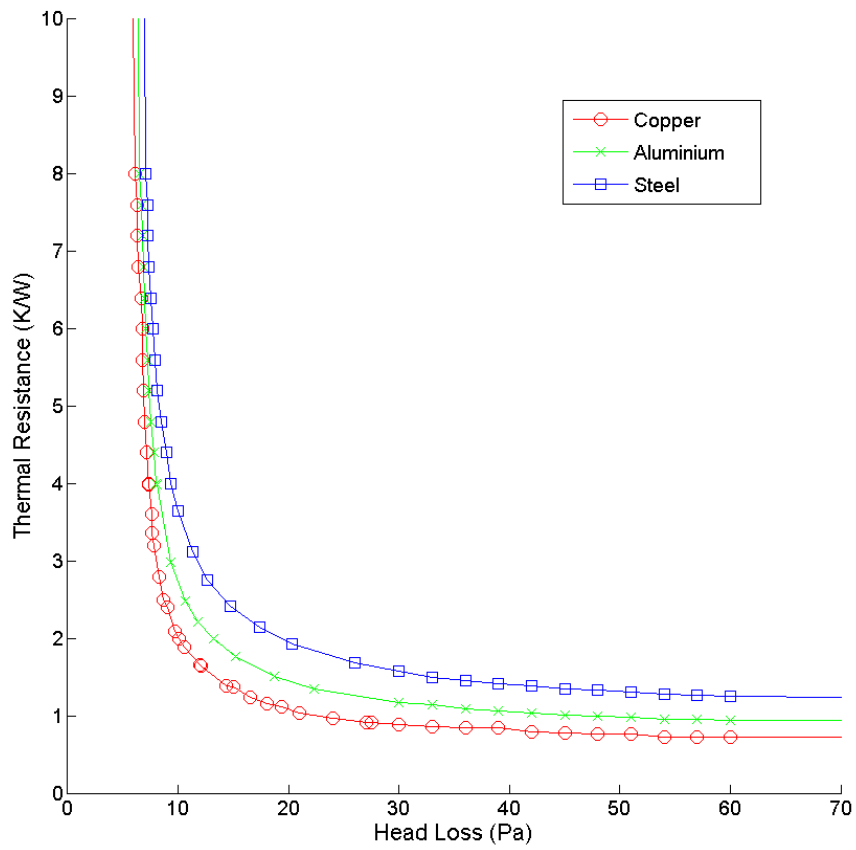


Figure 2. Pareto front of the different optimal solutions

How Much Forcing is Necessary to Let the Results of Particle Swarms Converge?

Bernd Bassimir, Manuel Schmitt, Rolf Wanka

Department of Computer Science, University of Erlangen-Nuremberg, Germany
`bernd.bassimir@fau.de`, `{manuel.schmitt,rolf.wanka}@cs.fau.de`

Abstract. In order to improve the behavior of Particle Swarm Optimization (PSO), the classical method is often extended by additional operations. Here, we are interested in how much “PSO” remains in this case, and how often the extension takes over the computation. We study the variant of PSO that applies random velocities (then called *forced moves*) as soon as the so-called *potential* of the swarm falls below a certain bound. We show experimentally that the number of iterations the swarm actually deviates from the classical PSO behavior is small as long as the particles are sufficiently far away from any local optimum. As soon as the swarm comes close to a local optimum, the number of forced moves increases significantly and approaches a value that depends on the swarm size and the problem dimension, but not on the actual fitness function, an observation that can be used as a stopping criterion. Additionally, we provide an explanation for the observed phenomenon in terms of the swarms potential.

1 Introduction

In the past few years Particle Swarm Optimization (PSO) has received increased attention because it can be easily implemented and adapted to the users’ applications. Unfortunately, the results are not always as good as wanted. Hence, many authors present changes in the original, “plain,” or classical PSO scheme (for exact definitions, see Sec. 2) in order to improve the quality of the returned solution. In the following, some of these PSO variants are mentioned.

In [7], van den Bergh/Engelbrecht introduce a PSO variant where the particles are allowed to count the number of times they improve the global attractor and to use this information.

In [5], a discrete variant of PSO for the Traveling Salesperson Problem is presented, where the PSO mechanism is enhanced by additional k -OPT-based intermediate phases. However, as experiments turn out, the quality of the solutions this algorithm returns does not significantly change when the PSO is completely turned off.

The phenomenon of premature stagnation, i. e., the convergence of the swarm to a non-optimal solution, has been addressed by Lehre/Witt [2]. To overcome such stagnation, they propose *Noisy PSO* that adds a “noise” term to the velocity at every move. The authors prove that for the Noisy PSO started on a

certain 1-dimensional function the first hitting time of the ε -neighborhood of the global optimum is finite.

As proved in [3], premature stagnation of classical PSO does not occur when the search space is 1-dimensional. I. e., in the 1-dimensional case, PSO provably finds a local optimum. Furthermore, [3] shows a similar result for a slightly modified PSO in the more general D -dimensional case. This modified PSO assigns a small random velocity only if the so-called *potential* of the swarm falls below a certain (small) bound. Such moves are called *forced*. This modified PSO provably finds a local optimum.

The goal of this paper is to determine, how much the modified algorithm relies on the forced moves, i. e., how often the modification actually is applied. We present experiments, implying that the forced moves will only have a small impact on the overall behavior of the swarm, as long as it is far away from any local optimum. Furthermore, our experiments show that the number of forced moves does not only increase significantly when the distance to the next local optimum falls below a certain bound, but that additionally the number of forced moves performed near a local optimum is independent of the fitness function. Therefore, the concentration of forced moves can act as a stopping criterion. Additionally, we give experiments that show the dependencies of this concentration, i. e., how it is influenced by the swarm size and the problem dimension.

2 Definitions

Definition 1 (Classical PSO Algorithm) *A swarm \mathcal{S} of N particles moves through the D -dimensional search space \mathbb{R}^D . Each particle $n \in \mathcal{S}$ consists of a position $X^n \in \mathbb{R}^D$, a velocity $V^n \in \mathbb{R}^D$ and a local attractor $L^n \in \mathbb{R}^D$, storing the best position particle n has visited so far. Additionally, the swarm shares information via the global attractor $G \in \mathbb{R}^D$, describing the best point any particle has visited so far, i. e., as soon as a particle has performed its move¹, it possibly updates the global attractor immediately.*

The actual movement of the swarm is governed by the following movement equations where $\chi, c_1, c_2 \in \mathbb{R}^+$ are some positive constants to be fixed later and r and s are drawn u. a. r. from $[0, 1]^D$.

$$\begin{aligned} V^n &:= \chi \cdot V^n + c_1 \cdot r \odot (L^n - X^n) + c_2 \cdot s \odot (G - X^n) \\ X^n &:= X^n + V^n \end{aligned}$$

Here, \odot denotes entrywise multiplication (Hadamard product). The application of the equation on particle n is called the move of n . When all particles have executed their moves, the swarm has executed one iteration.

Now we define a swarm's potential measuring how close it is to convergence, i. e., we describe a measure for its movement. A swarm with high potential should

¹ the particles' moves are executed sequentially, so there is some arbitrary order of the particles

be more likely to reach search points far away from the current global attractor, while the potential of a converging swarm approaches 0. These considerations lead to the following definition [4]:

Definition 2 (Potential) For $d \in \{1, \dots, D\}$, the potential of swarm \mathcal{S} in dimension d is Φ_d with $\Phi_d := \sum_{n=1}^N (|V_d^n| + |G_d - X_d^n|)$. The total potential of \mathcal{S} is $\Phi = (\Phi_1, \dots, \Phi_D)$.

The current total potential of a swarm has a portion in every dimension. Between two different dimensions, the potential may differ much, and “moving” potential from one dimension to another is not possible. On the other hand, along the same dimension the particles influence each other and can transfer potential from one to the other. This is the reason why there is no potential of individual particles.

To address the phenomenon of stagnation, we modify the movement equations from Definition 1 as follows [3]:

Definition 3 (Modified PSO) The modified movement of the swarm is governed by the following movement equations where $\chi, c_1, c_2, \delta \in \mathbb{R}^+$ are some positive constants to be fixed later and r and s are drawn u. a. r. from $[0, 1]^D$.

$$\begin{aligned} V^n &:= \begin{cases} (2 \cdot r - 1) \cdot \delta, & \text{if } \forall d \in \{1, \dots, D\} : |V_d^n| + |G_d - X_d^n| < \delta \\ \chi \cdot V^n + c_1 \cdot r \odot (L^n - X^n) + c_2 \cdot s \odot (G - X^n), & \text{otherwise,} \end{cases} \quad (1) \\ X^n &:= X^n + V^n. \end{aligned}$$

If the first case of (1) applies for a particle, we call its move forced. An iteration of the swarm is called forced if at least one particle performs a forced move.

Algorithm 1 below provides an overview over the modified PSO. The introduction of forced moves guarantees that the swarm does not converge to a non-optimal point, but finds a local optimum [3].

3 Number of Forced Iterations

In the following experiments, we tested the modified PSO algorithm with the fixed parameters $\chi = 0.729, c_1 = c_2 = 1.49$ as suggested in [1]. Additionally, we set the swarm size $N = 3$, the problem dimension $D = 30$ and the parameter $\delta = 10^{-6}$. The total number of iterations was set to 4.000.000 and for every period of 2.000 iterations, we counted the forced iterations. As fitness functions, we used selected benchmarks from [6], namely SPHERE (F_1), SCHWEFEL 1.2 (F_2), (unrotated) H. C. ELLIPTIC (F_3), ROSENBROCK (F_6), RASTRIGIN (F_9) and SCHWEFEL 2.13 (F_{12}). The results can be seen in Figure 1. Initially, the positions were uniformly and independently distributed over $[-100, 100]^D$ and the velocities were uniformly and independently distributed over $[0, 1]^D$. To measure the quality of the solution found at each time, the gradient is calculated at the

Algorithm 1: Modified PSO

```
    output:  $G \in \mathbb{R}^D$ 
1  for  $n = 1 \rightarrow N$  do
2    Initialize  $X^n$  randomly; Initialize  $V^n$  randomly;
3    Initialize  $L^n := X^n$ ; Initialize  $G := \operatorname{argmin}\{L^n\}$ ;
4  repeat
5    for  $n = 1 \rightarrow N$  do
6      if  $\forall d \in \{1, \dots, D\} : |V_d^n| + |G_d - X_d^n| < \delta$  then
7         $V^n := (2 \cdot r - 1) \cdot \delta$ ;
8      else
9         $V^n := \chi \cdot V^n + c_1 \cdot r \odot (L^n - X^n) + c_2 \cdot s \odot (G - X^n)$ ;
10     if  $f(X^n) \leq f(L^n)$  then  $L^n := X^n$ ;
11     if  $f(X^n) \leq f(G)$  then  $G := X^n$ ;
12 until termination criterion met;
```

end of each interval. In each run and for every examined fitness function, we can observe the following behavior. In the beginning of the optimization process, the number of forced iterations is low. When the swarm reaches (the neighborhood of) a local optimum, this number will rise and then begin to stagnate around a certain value. While the time when that happens may depend on the fitness function, one can clearly see that this value does not. Note that the point in time when the particles come close to a local optimum and the number of forced iterations increases is random. In case of SCHWEFEL 2.13, the point has a comparatively high variance, so the smaller increase of the respective curve in Figure 1 results from averaging over the 100 runs rather than generally smaller increases of the single runs on this function. As seen in Figure 1, the number of forced iterations is low in the beginning of the algorithm. This is due to the nature of the modification made. Whenever the swarm begins to converge to a point that is no local optimum the modification applies and enables the swarm to search for a new, promising direction. For that purpose only a few forced iterations are necessary. When the swarm reaches a local optimum, the modification is also applied and the swarm will again try to find a new direction. However, since every direction yields worsenings, the swarm will never switch back to standard swarm behavior for longer than a few iterations. From Figure 1, one can see that the number of forced moves at that point will stay around a certain and comparatively high value. An explanation is that the particles will now be gathered in a neighborhood with radius of about δ around the global attractor. At that point, the global attractor is very close to the local optimum, so the swarm's potential is small. As known from, e.g., [1], the unmodified PSO converges and therefore loses potential. Consequently, as soon as the modified PSO has enough potential to apply the classical movement equations, after some iterations it is again in a situation where its moves are forced. Therefore the stochastic process of the particle swarm becomes a stationary process. The number of forced iterations

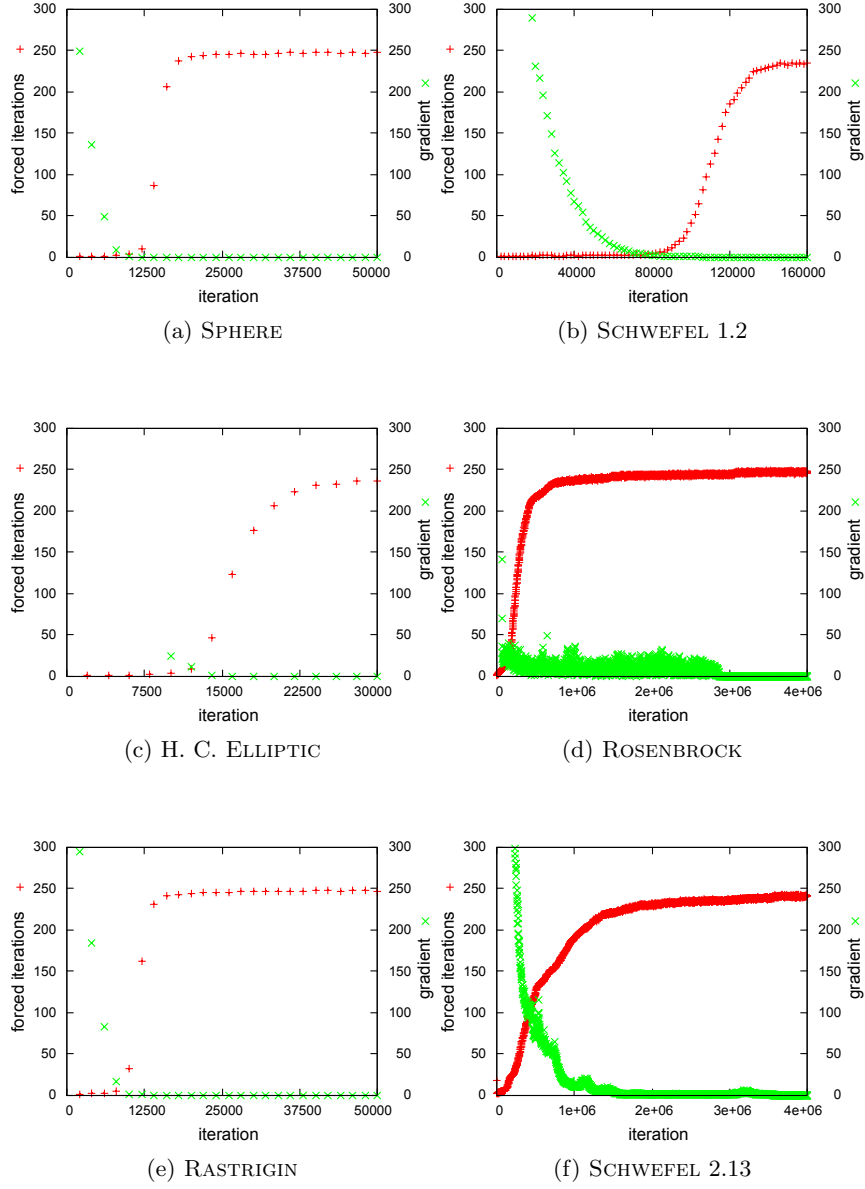


Fig.1: Forced iteration count and gradient for various fitness functions. Each point represents the average over 100 different runs. The forced iteration count is the sum over 2,000 iterations and the gradient is calculated at the global attractor every 2,000 iterations.

under this stationary distribution will in the following be called the *stagnation value*. The stagnation value is independent of the fitness function, because the changes of the attractors are the only influence the function has on the movement of the particles. So the smaller the attractors' movement gets, the less relevant is the actual function for the behavior of the swarm. Note that the time when the particles come close to a local optimum decreases when the swarm size is increased. In particular, with a larger swarm size, this time is less than 2000. Consequently, we chose a very low number of particles to highlight the difference between the optimization phase, when the behavior of the swarm is mostly governed by the standard movement equations, and the stagnation phase, when a large share of the iterations is forced.

4 Dependencies of the Stagnation Value

In Section 3 we have seen that the value at which the number of forced iterations stagnates is independent of the fitness function. In this section we will give a further insight into this phenomenon. We provide experimental results pointing out that there are two dependencies for this value, namely the search space dimension and the swarm size. Figure 2 shows how the stagnation value changes when these parameters are varied. Each point is the arithmetic mean over the last 50 values of the forced iteration number, i. e., the values are obtained from the last 100.000 iterations. Each run in Figure 2a consists of 500.000 iterations and each run in Figure 2b of 200.000 iterations. The chosen fitness function is the SPHERE.

In Figure 2a, one can see that the value at which the number of forced iterations stagnates decreases exponentially as the number of dimensions increases. To explain this behavior, we have to look at the probability of the occurrence of a forced move. In Equation (1), one can see that a forced move will occur when the sum of the absolute value of the velocity and the distance of the particle to the global attractor fall below a certain bound in every dimension. As soon as the swarm is sufficiently close to a local optimum, such that updates of attractors are rare and the differences between the old values and the updated values are small, the dimensions are almost independent of each other. The probability of the necessary condition for a forced move, as specified in (1) of Definition 3, to be fulfilled in a given dimension does not depend on the actual dimension. This leads to the following equation describing the probability of a forced move for a given particle n .

$$\Pr(\text{forced_move}_n) \approx \Pr(|V_1^n| + |G_1 - X_1^n| < \delta)^D$$

Additionally, the stagnation value depends on the swarm size. As specified before, we call an iteration forced when at least one particle performs a forced move, i. e., the following equation holds:

$$\Pr(\text{forced_iteration}) = \Pr\left(\bigcup_{n=1}^N \text{forced_move}_n\right)$$

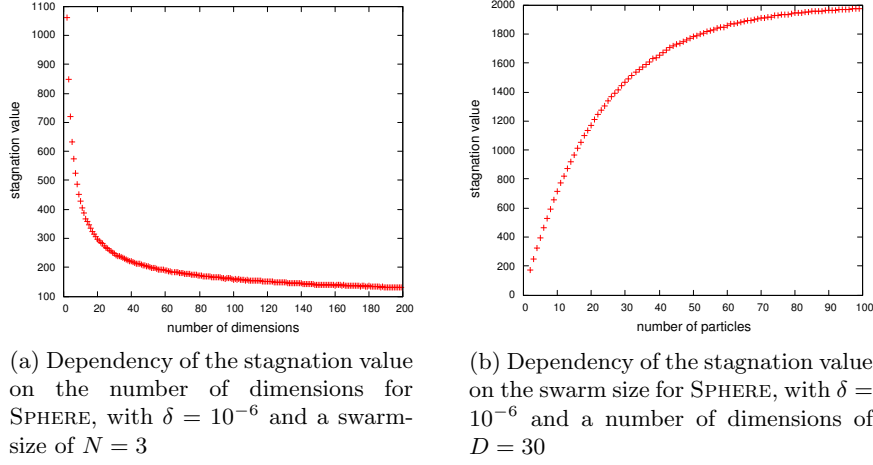


Fig. 2: Influence of dimension number and swarm size on the stagnation value

Figure 2b shows the obtained stagnation value for a fixed search space dimension of $D = 30$ and swarm sizes from 2 to 100. As one can see, the increase of the stagnation value is almost linear for small swarm sizes. An explanation for this phenomenon is the following: At the time of the stagnation, the global attractor will be almost constant. As a direct consequence of this, the particles are moving almost independent from all other particles. With a low number of particles used, the resulting value is nearly the sum over the probabilities that particle n performs a forced move for all particles. As this number rises, it can be observed that this value converges towards 2000. At this point, the probability of more than one forced move during the same iteration rises and has to be taken into account. This convergence to 2000 comes from the stagnation value being calculated as the sum over 2000 iterations. Given this two dependencies and the results shown in Section 3, we conclude that the stagnation value of the number of forced iterations is a function $F_{\max}(N, D)$ that is independent of the fitness function. To compute a result of this function given the two parameters, one can first run the PSO algorithm for a simple function like SPHERE. The obtained stagnation value can then be applied as a stopping criterion in the PSO algorithm that computes the desired fitness function with the same values for N and D .

5 Conclusion

This paper focused on a modified PSO variant, in which the velocity update is done uniformly over a small interval whenever the swarm reaches a state close to convergence. We gave experiments suggesting that the influence of this modification is relatively small as long as the swarm is far away from any local optimum. Additionally, the experiments have clearly shown that the number of forced

iterations reaches a stationary distribution when a local optimum is reached. Under that stationary distribution, the number of forced iterations is orders of magnitude higher than the number of forced iterations performed during the optimization process. The magnitude was further studied and experiments were given to show its dependencies. Therefore, this concentration of forced iterations can act as a stopping criterion.

References

1. M. Clerc and J. Kennedy. The particle swarm – explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6:58–73, 2002. doi:10.1109/4235.985692.
2. P. K. Lehre and C. Witt. Finite first hitting time versus stochastic convergence in particle swarm optimisation, 2011. arXiv:1105.5540.
3. M. Schmitt and R. Wanka. Particle swarm optimization almost surely finds local optima. In *Proc. 15th Genetic and Evolutionary Computation Conference (GECCO)*, pages 1629–1636, 2013. doi:10.1145/2463372.2463563.
4. M. Schmitt and R. Wanka. Particles prefer walking along the axes: Experimental insights into the behavior of a particle swarm. In *Companion of Proc. 15th Genetic and Evolutionary Computation Conference (GECCO)*, pages 17–18, 2013. doi:10.1145/2464576.2464583.
5. X. H. Shi, Y. C. Liang, H. P. Leeb, C. Lu, and Q. Wang. Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, (103):169–176, 2007. doi:10.1016/j.ipl.2007.03.010.
6. 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, KanGAL Report Number 2005005 (Kanpur Genetic Algorithms Laboratory, IIT Kanpur), 2005.
7. F. van den Bergh and A. P. Engelbrecht. A new locally convergent particle swarm optimiser. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics (SMC)*, volume 3, pages 94–99, 2002. doi:10.1109/ICSMC.2002.1176018.

Title : Smooth trajectory planning for robot using particle swarm optimization.

Riad Menasri, Hamouche Oulhadj, Boubaker Daachi, Amir Nakib, Patrick Siarry

University Paris-Est Creteil, LISSI Laboratory, 122 rue Paul Armangot, 94400 Vitry Sur Seine, France

In this work, we deal with a class of problem of trajectory planning taking into account the smoothness of the trajectory. We assume that we have a set of positions in which the robot must pass. These positions are not assigned in the time axis. This kind of result can be found in many works of trajectory planning. Thus, this result is not complete in the sense that we don't have the complete trajectory and, we do not have any idea about the whole time to move.

In this work, we propose a formulation of this problem, where the total length of the trajectory and the total time to move from the initial to the final position are minimized simultaneously. In order to have a good result and avoid abrupt movement, we should ensure the smoothness of the trajectory not only at the position level but also at the velocity and the acceleration level. Thus, the position function must be at least two times differentiable. In our case, we use a polynomial function. These functions have the advantages that we can modify the data easily by multiplying or dividing the coefficients by the same factor. Thus, we formulate this problem as a constraint optimization problem. To resolve it, we adapt the usual particle swarm algorithm to our problem and we show its efficiency by simulation.

PSO: A Powerful and Fast Intelligence Optimization method In Processing of Passive Geophysical Data.

Rashed Poormirzaee^{1*}, Rasoul Hamidzadeh Moghadam¹, Ahmad Zarean²

¹Sahand University of Technology, Tabriz, Iran

Rashed.Poormirzaee@gmail.com

rhamidzm@yahoo.com

²Islamic Azad University, Shabestar Branch, Tabriz, Iran

zarean@gmail.com

Abstract

With the advancement of data optimization algorithms, fast and easier approaches can be conducted for processing of geophysical data. The current study intends to discuss applying Particle Swarm Optimization (PSO) algorithm for processing of passive geophysical data. In this study we investigate the reliability and efficiency of PSO algorithm in inversion, as a main stage of processing of Refraction Microteremor (ReMi) data. ReMi method as a passive approach is widely used in near surface studies. So, first we developed PSO code in Matlab for inversion of ReMi data and then the efficiency of proposed algorithm investigated by inversion a noise-free and a noise-corrupted synthetic data set. The findings show that PSO algorithm is a powerful, fast and easy for inversion of passive geophysical data.

Keywords: PSO, Processing, Geophysical data.

1 Introduction

By passing the time, new angles of technology in different fields like computer science and information processing have been identified and human beings are getting more and more curious to make interdisciplinary application of emerging technologies. One of the fields of studies that had been a huge effect on engineering science progress is optimization techniques. In general, optimization algorithms can be divided into two categories: deterministic algorithms, and stochastic algorithms. Deterministic algorithms follow a rigorous procedure and its path and values of both design variables and the functions are repeatable. Most conventional or classic algorithms are deterministic (e.g. Simplex method). On the other hand, the stochastic algorithms always have some randomness. For stochastic algorithms, we have in general two types: heuristic and metaheuristic, though their difference is small. Further development over the heuristic algorithms is the so-called metaheuristic algorithms. They generally perform better than simple heuristics. In addition, all metaheuristic algorithms use certain tradeoff of randomization and local search [1]. PSO algorithm is one of the global optimization methods that belong to group of metaheuristic searching algorithms. Inverse theory is an organized set of mathematical techniques for reducing data to obtain useful information about the physical world on the basis of inference drawn from observation [2]. Inversion of geophysical data consists of operating directly on those data so as to generate a view of the structure which causes them [3]. To solve an inverse problem design of three step is vital : i)parameterization of the system(i.e., discovery of a minimal set of model parameters whose values completely character-

* Sahand University of Technology, Tabriz, Iran.

ize the system), ii) forward modeling (i.e., discovery of the physical laws allowing us, for given values of the model parameters, to make prediction on the results of measurements on some observable parameters), iii) inverse modeling (i.e. use of the actual results of some measurements of the observable parameters to infer the actual values of the model parameters) [4]. Most geophysical inversion methods are based on linearized techniques to estimate the parameters of model in an iterative manner; i.e., using local optimization algorithm to modify a starting model that user defined it. At each iteration, a better estimate of the model is calculated by linearizing the problem and the best solution, minimizing a misfit function, is obtained after a few iterations [5]. Since most geophysical inverse problems are nonlinear, hence have non-linear misfit functions (e.g. RMS travel-time error), the solution is quite often trapped to local minima during the application of local optimization methods. As a result, their success depends on the initial model to the “true” global-minimum solution. But global optimization algorithms include the ability to produce solutions independent on the initial model, to explore the model space in more detail and, thus, a better chance to find the “true” global minimum solution [6, 7]. So, by finding of global minimum of misfit function the best solution of problem could be reach. Therefore using an optimization algorithm in inversion of geophysical data that satisfy above condition for finding best solution is an important issue. In geophysical surveys, there have recently been emerged several significant PSO applications. The PSO on a multilayered 1D vertical electric sounding (VES), induced polarization (IP), magnetotelluric (MT) methods both synthetic and field data have successfully been carried out by Shaw and Srivastava (2007)[8]. Naudet et al. (2008) [9] have studied water table estimation using the PSO on self-potential (SP) data. Inversion of a VES data for environmental applications by the GPSO algorithm has successfully been performed by Fernández Martínez and García Gonzalo (2008)[10]. Yuan et al. (2009)[11] have demonstrated PSO worked on seismic wavelet estimation and gravity anomalies as well. Also they concluded that the PSO inversion method have the attributes of higher convergence speed and accuracy than conventional GA and SA methods. Fernández Martínez et al. (2010) [12] present the application of a whole family of PSO algorithms to the analysis and solution of a VES inverse problem associated with a seawater intrusion in a coastal aquifer in southern Spain. Pekşen et al. (2011) [13] by application of PSO algorithm have inverted self-potential (SP) data. In recent years Rayleigh waves have attracted the interest of a constantly increasing number of researchers from different disciplines for a wide range of applications [14]. Once Rayleigh wave dispersion curve is properly identified, its inversion is the key point to obtain a reliable near-surface S-wave velocity profile. X. Song et al. (2012)[15] proposed Rayleigh wave dispersion curve inversion scheme based on particle swarm optimization. PSO is a novel and powerful technique in geophysical data interpretation. However, no attempts have been made to investigate the reliability of PSO algorithm in inversion of ReMi data as a passive geophysical method with a limit range of frequency. In this study, we demonstrate a PSO application on ReMi data inversion. To evaluate efficiency and stability of PSO to invert of ReMi data, we first developed PSO code in Matlab and then investigated on a synthetic model. Also in order to solve the forward modeling and estimate the theoretical dispersion curve the code based on the matrix algorithm developed by Herrmann (1987) [16] was used. The root mean square (RMS) error is considered as the objective function (OBF) and calculated for the entire set of random velocity and thickness models of each particle. Results from synthetic

data demonstrate that particle swarm optimization can be used for processing of ReMi data.

2 ReMi Method

The ReMi method, introduced by Louie [17], has widely been used to determine shear wave velocity profiles using ambient noise recordings. The ReMi method provides an effective and efficient means to obtain general information about large volumes of the subsurface in one dimension per setup, where appropriate setup length is related to desire the depth of investigation [18]. The determination of underground shear-wave velocity (V_s) is a very important issue in geotechnical and earthquake engineering because of its direct relationship with the stress and strain properties of soils [19]. There are a number of geophysical techniques commonly applied in near surface study to the task of estimating the V_s structure at a site (e.g. SASW, MASW, reflection, refraction, borehole seismics and array Microtremor). Urban conditions, such as existing underground facilities and ambient noise due to cultural activity, restrict the general application of conventional geophysical techniques [20]. The refraction microtremor method combines the urban utility and ease of microtremor array techniques with the operational simplicity of the SASW technique and the shallow accuracy of the MASW technique. By recording urban microtremor on a linear array of a large number of lightweight seismometers, the method achieves fast and easy field data collection without any need for the time-consuming heavy source required for SASW and MASW work [17]. As SASW technique ReMi is based on obtaining the dispersion curve of the Rayleigh waves, but in this case using ambient seismic noise or Microtremors. Standard refraction equipment is used, usually a linear array of 12 or more geophones sensors channels that register surface waves at low frequency [19]. The common seismic refraction equipment can yield accurate surface-wave dispersion information from microtremor noise. Configurations of 12 to 48 single vertical, 8-12 Hz exploration geophones can give surface-wave phase velocities at frequencies as low as 2 Hz, and as high as 26 Hz. This range is appropriate for constraining shear velocity profiles from the surface to 100 m depths [17]. After J.Louei (2001) [17], reliability and accuracy of ReMi method was investigated in different case studies. Rucker ML. (2003)[18] Applied the ReMi shear wave technique to geotechnical characterization and several geotechnical applications have been presented in his publication. Scott B.J. and et al. (2004) [20] performed measurement of shear-wave velocity to 30 m depth (V_{s30}) for hazard assessment of Reno basin. They were successful in obtaining detailed shallow shear-wave velocity transect across an entire urban basin with minimum effort.

3 PSO Algorithm

Particle swarm optimization (PSO) is a stochastic evolutionary computation technique for optimization in many different engineering fields, which is inspired by the social behavior of individuals (called particles) in groups in nature, such as a flock (swarm) of birds searching for food [21,22]. Particle swarm optimization may have some similarities with genetic algorithms and ant algorithms, but it is much simpler because it does not use mutation/crossover operators or pheromone.

This algorithm searches the space of an objective function by adjusting the trajectories of individual agents, called particles, as these trajectories form piecewise paths in a quasi-stochastic manner [1]. The particles are moving towards promising regions of the search space by exploiting information springing from their own experience during the search, as well as the experience of other particles. For this purpose, a separate memory is used where each particle stores the best position (x_i^*) it has ever visited in the search space. The best position of each particle experience comprised to other ones and then the best position, which belongs to minimum of misfit function, selected as the global best (g^*). This procedure (i.e. finding x_i^*, g^*) repeated for certain iteration. Finally the best global g^* is determined as the optimum solution. The movement of particles is schematically represented in Figure 1.

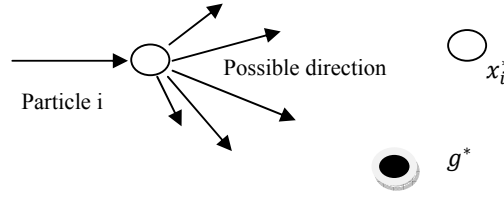


Fig.1. Schematic representation of the motion of a particle in PSO, moving towards the global best g^* and the current best x_i^* for each particle i [1].

4 PSO for inversion of ReMi data

Because conventional seismic equipment is used to record data, and ambient noise is used as a seismic source, the ReMi method is less costly, faster and more convenient than borehole methods and other surface seismic methods, such as SASW and MASW used to determine shear-wave profiles. This wide variety of applications leads to an equally wide variety of field survey methods and associated interpretation techniques. So, it seems that for accurate interpretation the processing of ReMi data and inversion of this data are an important issue. The recent geophysical literature includes many works on development and application of inversion techniques a few literatures there are that investigate the inversion of surface waves based data by use of metaheuristic methods. In this study we proposed new code using PSO algorithm with to aims of the inversion of Rayleigh wave data for study of near surface. This code is easy and fast also allows the user to inclusion of a priori information on the different parameters. The algorithm that used for PSO for inversion of ReMi data includes steps as follows:

Step1: Initial location x_i and velocity v_i of n particle:

Step2: Finding the x_i^* and g^* :

The root-mean-square (RMS) misfit between the observed and calculated value are defined as the object function (OBF) according to the following equation:

$$OBF = \sqrt{\frac{\sum_{j=1}^{n_p} (v^{obs} - v^{cal})^2}{n_p}} \quad (1)$$

Where n_p is the number of samples, v^{obs} is the observed velocity and v^{cal} is the calculated velocity.

According to Eq.1 misfit is calculated and then the particle which have the least misfit imported to g^* as a new value. The first position of each par-

ticle inclusion to x_i^* in first iteration. For next iteration the position of particle that have least misfit inclusion to the best position of it (i.e. x_i^*).

Step3: While maximum number of cycles has not been reach do:

- Update velocity of each particles (according to Eq.2):

$$v_i^{t+1} = \chi[v_i^t + \alpha\epsilon_1\odot(g^* - x_i^t) + \beta\epsilon_2\odot(x_i^* - x_i^t)] \quad (2)$$
- Update position of each particles (according to Eq.3):

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3)$$
- Maintain the particles within the search space, $[x_j^L, x_j^U]$, in case they go beyond their boundaries.
- Mutation.
- For new position of all particles calculate x_i^* according to step 2.
- Calculate the least misfit and inclusion correspond particle to g^* .

Step4: End while

Step5: Output the final result g^* .

5 Synthetic dataset

The method was test on a free-noise synthetic case (model A). Then we add 10 percent noise to the synthetic model. For each model the PSO algorithm with different parameters was run. So, For sake of brevity for model A just the best mean obtained model is represented. The particle and maximum of iteration number are 60 and 50 respectively. We used a simple earth model (homogeneous single layer overlying a half-space), model A. The Table 1 shows the model A and the search space of algorithm.

Table 1. Parameters of the synthetic model A and search space.

layer	Vp(m/s)	Vs(m/s)	Poisson	H(m)	search space	
					Vs(m/s)	H(m)
1	400	200	0.45	8	100-300	4-12
2	600	400	0.25	half space	200- 600	-

After running the PSO code with different parameters (inertia weight, cognitive and social coefficients and constriction factor), the results show that the best estimate of velocities of layers (Vs) and the thickness (H) obtained by parameters as Table 2. Also the model A (bold line) and mean estimate value (red points) are depicted in figure2. Also the results for model A with 10% noise are represented in Table 3 and figure 3.

Table 2. Mean model obtained from the model A.

parameters	True	Estimated	Related Error (%)	Standard deviation	PSO parameters
Vs1(m/s)	200	198	1.0	1.32	$\alpha=1.3$ $\beta=2.8$ $w=0.729$
Vs2(m/s)	400	388	3.0	9.62	
H1(m)	8	7.9	1.3	0.38	

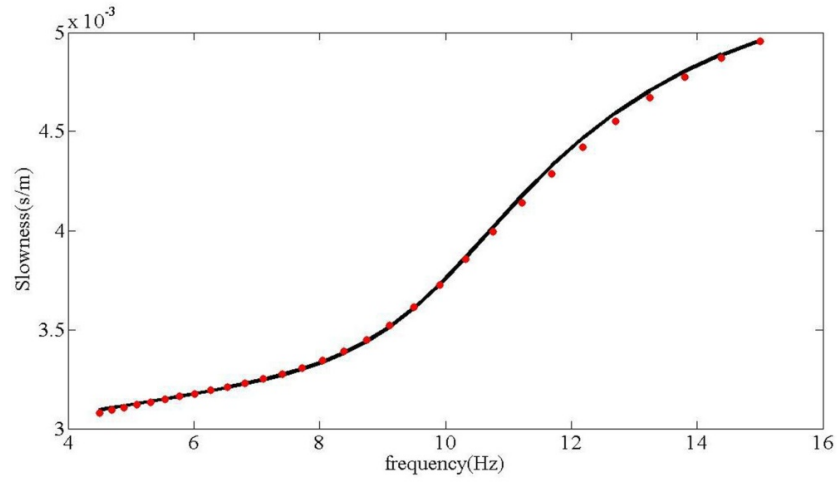


Fig. 2. model A (bold line) and mean estimated value(red points).

Table 3. Mean model obtained from the model A with 10% noise.

parameters	True	Estimated	Related Error (%)	Standard deviation	PSO parameters
Vs1(m/s)	200	197	1.5	3.08	$\alpha=1.3$
Vs2(m/s)	400	386	3.5	18.00	$\beta=2.8$
H1(m)	8	7.7	3.4	0.41	$w=0.729$

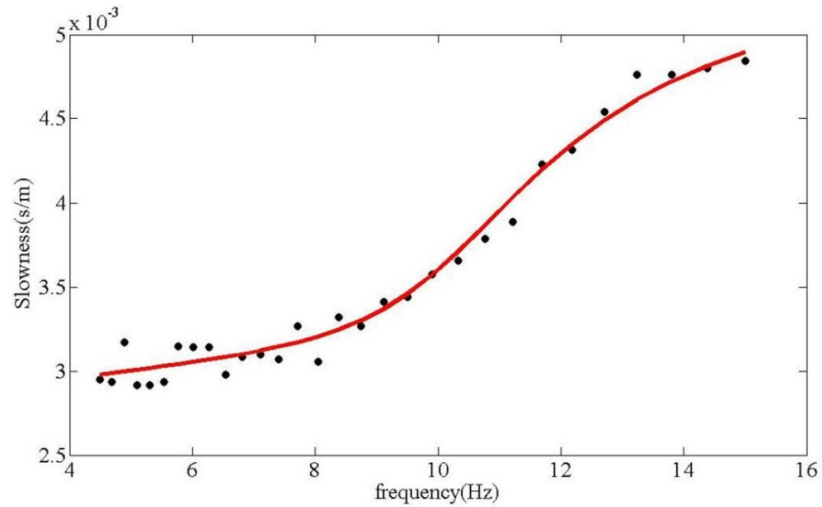


Fig. 3. model A with 10% noise (black points) and mean estimated value (red bold line).

6 Conclusion

ReMi method, have been used increasingly as an appealing tool for surveys of near surface shear wave velocity. The determination of underground shear-wave velocity is a very important issue in geotechnical and earthquake engineering because of its direct relationship with the stress and strain properties of soils. The ReMi data acquisition is easier and more time efficient, requiring less equipment than other geophysical methods. However, inversion, as a main stage in processing of ReMi data, is challenging for most local-search methods due to its high nonlinearity and to its multimodality. With the advancement of computer science, optimization algorithms and intelligence strategies, fast and easier approaches can be conducted for inversion of geophysical data with high nonlineari-

ty nature. PSO is a global optimization strategy that simulates the social behavior observed in a flock (swarm) of birds searching for food. A simple search strategy in PSO guides the algorithm toward the best solution by updating of the cognitive and social parameters. In this study we implemented PSO algorithm for inversion of ReMi data. The coding for invert ReMi data was done in Matlab. The purposed inversion algorithm ran on a synthetic model and added 10% noise. Between different proposed parameters of PSO (i.e. α, β, χ), the values (1.3, 2.8, and 0.729) give the best estimate. The great advantages of PSO inversion code are that it is fast and easy to implement and there are few parameters to adjust. Flexibility of implemented inversion code is its trait. The results proved the ability and reliability of PSO in inversion of passive geophysical data, with a limit range of frequency, for achieving to a correct model of 1D velocity model with an acceptable misfit and convergence speed.

Reference

1. Yang, X.S.: Engineering Optimization An Introduction with Metaheuristic Applications. John Wiley & Sons, pp.19-22 (2010)
2. Menke, W.: Geophysical Data Analysis: Discrete Inverse Theory. International Geophysics series, Revised Edition, volume 45, pp.10-12(1989)
3. Vozoff, K., Jupp, D. L. B.: Joint Inversion of Geophysical Data. Geophysic. J. R. astr. Soc. 42,977-991 (1975)
4. Tarantola, A.: Inverse problem theory and methods for model parameter estimation. Society for Industrial and Applied Mathematics, ISBN 0-89871-572-5 (pbk.), pp.1-4. (2005)
5. Wathelet, M., Jongmans, D. and Ohrnberger, M.: Surface wave inversion using a direct search algorithm and its application to ambient vibration measurements. Near Surface Geophysics, 2, 211-221(2004)
6. M. K. Sen and P. L. Stoffa: Global optimization methods in geophysical inversion. Elsevier Science(1995)
7. Soupios, P. , Akca, I., Mpogiatis, P., Basokur, A., T, Papazachos , C.: Applications of hybrid genetic algorithms in seismic tomography. J. of Applied Geophysics 75, 479–489(2011)
8. Shaw, R., Srivastava, S.: Particle swarm optimization: a new tool to invert geophysical data. Geophysics 72 (2), F75–F83(2007)
9. Naudet, V., Fernández Martínez, J.L., García Gonzalo, E., Fernández Alvarez: Estimation of water table from self-potential data using particle swarm. J.P. pp. 1203–1207(2008)
10. Fernández Martínez, J.L., García Gonzalo, E.: The generalized PSO: a new door to PSO evolution. J. for Artificial Evolution and Applications, pp.1–15(2008)
11. Yuana , S., Shangxu, W., Nan, N.: Swarm intelligence optimization and its application in geophysical data inversion. Applied Geophysics. Vol.6, No.2, P. 166 – 174(2009)
12. Fernández Martínez, J.L., García Gonzalo, E., Fernández Álvarez, J.P., Kuzma, H.A., Menéndez Pérez, C.O.: PSO: a powerful algorithm to solve geophysical inverse problems: application to a 1D-DC resistivity case. J. of Applied Geophysics 71, 13–25(2010)
13. Pekşen, E., Yas, T.A., Kayman, Y., Özkan, C.: Application of particle swarm optimization on self-potential data. J. of Applied Geophysics. 75 (2), 305–318(2011)
14. Coccia, S. , Del Gaudio, V. , Venisti, N., Wasowski, N.: Application of Refraction Microtremor (ReMi) technique for determination of 1-D shear wave velocity in a landslide area. J. of Applied Geophysics. 71, 71–89(2010)
15. Song, X., H. , Tang, T., Lv, X., Fang, H., Gu, H.: Application of particle swarm optimization to interpret Rayleigh wave dispersion curves. J. of Applied Geophysics. 84,1–13(2012)
16. Herrmann, R.B.: Computer Programs in Seismology. St Louis University (1987)
17. Louie, J.N.: Faster, better: shear wave velocity to 100 meters depth from refraction Microtremor arrays. Bull. Seism. Soc. Am., v. 91, p. 347–364 (2001)
18. Rucker, M.,L.: Applying the refraction microtremor (ReMi) shear wave technique to geotechnical characterization. In: Proceedings of the third international conference on the application of geophysical methodologies and NDT to transportation and infrastructure, Orlando, Florida, December 8–12(2003)

19. Pérez-Santisteban, I., García-Mayordomo, J. , Muñoz Martín, A. , Carbó, A.: Comparison among SASW, ReMi and PS-logging techniques: Application to a railway embankment, *Journal of Applied Geophysics* 73,59–64(2011)
20. Scott, J., B., Clark, M., Rennie, T., Pancha, A., Park, H., Louie, J., N.: A Shallow Shear-Wave Velocity Transect across the Reno, Nevada, Area Basin. *Bulletin of the Seismological Society of America*. Vol. 94, No. 6, pp. 2222–2228(2004)
21. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. IEEE Press, Piscataway, NJ. pp: 1942–1948(1995)
22. Blum, C., Li, X.: Swarm Intelligence in Optimization. Natural Computing Series, Springer Berlin Heidelberg, pp: 43-85(2008)

Accelerated Particle Swarm Optimisation: Weather Networks Architecture Case Study

Redouane Boudjemaa
rboudjemaa@umbb.dz

University Mhamed Bougara of Boumerdes

Abstract. This work proposes an approach to the optimal placement of a weather radar network based on an accelerated particle swarm optimisation. Given a finite number of weather radars, a network is produced such that the geographical coverage area of the radars is maximised. By taking into account several constraints such as terrain blockage and radar beam elevation, the problem is formulated mathematically into an optimisation problem. By transforming the search space into a gridded system, a reduction in the number of possible combinations of radar networks is achieved making the problem manageable in size. Numerical results show a good convergence to an acceptable solutions based on the percentage of area covered. The proposed approach can serve as an analysis tool for a decision support system to assist meteorologist in the selection of future prime sites for the installation of weather radars.

Keywords: Accelerated PSO, Weather Radar Network, Optimal placement.

1 Introduction

Weather Radar Networks (WRN) have been commonly used in both the prevision and research of weather systems. An example of a collaborative radar network is the operational programme for the exchange of weather radar information (OPERA) which is a joint project of 28 European countries [5]. The OPERA network includes close to 160 operational radars with some 110 being Doppler radars. A difficult task in constructing these networks is determining adequate sitting sites of radars in order to meet certain conditions. As declared by Cole and Mass [1], the benefits of such a WRN is the ability to study regional weather phenomena and most importantly over landscapes with complex orography, such as mountainous regions Joss et al. [4]. Minciardi et al. [13] explain that contrary to a satellite observation of a weather system which only concentrate on the mass movement of cloud tops, the WRN can provide a more accurate measurement and real time data over a wide spread spatial region. This is due to the fact that the core of heavier precipitation lies at an altitude below one kilometre as pointed out by Wilson et al. [6]. The WRN provides also spatial distributed rainfall measurements, an important data for hydrology systems, in contrast to the rain gauges equipment which provides only a point measurement.

The introduction of small and cheap X-BAND radars to the WRN would provide a regional coverage at a sub 1 km height levels which is essential for the provision of fast and high resolution data. Such a reliable and accurate data would allow a real time update to weather forecast models resulting in the issue of important warnings of severe weather phenomena [10].

Limited literature exists in terms of optimisation methods applied to optimal placement of a weather radar (WR). The approach that is often used in previous works is the selection of probable radar sitting sites based on a number of pre-set criterion. A systematic and objective approach in the placement of the Next Generation Weather Radar (NEXRAD)(WSR-88D) Doppler radars was investigated in the work carried out by Leone et al [2].

An important breakthrough was achieved by Minciardi et al. [13] by establishing for the first time a mathematical well defined optimisation problem. The total geographical region was divided to 285 subregions and four different covering layers were taken into account. Out of the 107 WRs, twenty were in the process of installation while the sitting sites of the remaining ones were evaluated through an optimisation of an adapted weighted set-covering problem.

A more recent and advanced work in determining the placement of WRs is investigated by Kurdzo et al. [9]. Through the utilisation of a genetic algorithm (GA) a maximisation of the coverage area within a set physical boundary condition is achieved. They provides three different optimisation problems to illustrate the flexibility and scope of their algorithm.

Particle swarm optimization (PSO) [7] is a population-based stochastic search algorithm which has become very popular for solving different scientific and engineering problems. Contrary to evolutionary algorithms which are based on the concepts of natural evolution (competitiveness) where only the fittest can survive, PSO incorporates a cooperative approach in searching for a solution. Survival and fitness are simultaneously allowed to all individuals (particles) and the successful fitness of a particle is transmitted and reflected into the behaviour of its neighbours as presented by Kennedy [8].

PSO has been used extensively and with success in optimal placement problem in the field of power systems. Pookpant et al. [11] utilised a Binary PSO algorithm in an optimal placement of wind turbines within a wind farm with an objective to extract the maximum turbine power output with minimum investment cost.

Pradhan and Panda [12] optimised sensor node locations in a wireless sensors network using a multiobjective PSO. By conducting a sensitivity analysis of different parameters they showed that the multiobjective particle swarm optimization algorithm is a better candidate for solving the multiobjective problem of deploying the sensors.

In this paper we present an algorithm based on accelerated particle swarm optimisation (APSO) to locate suitable sites for the installation of WRs in a constrained geographical region. Given a number of available radars, the algorithm will place each and every one of them in a suitable geographical location maximising the total coverage area. Terrain blockage and radar beam elevation

above a 1 km altitude are taken into account in the evaluation of a proposed radar placement site. This algorithm assumes that all possible sites have the same cost of installation and maintenance.

2 Application of PSO to WRN optimisation

Particle swarm optimization (PSO) was developed by Kennedy and Eberhart in 1995 [7], based on swarm behaviour such as fish and bird schooling to a place with enough food or to seek safety. As described by Yang [14], the movement of each swarming particle is determined by a combination of a stochastic element and a deterministic element. Each particle moves randomly while at the same time it is attracted toward a global leading particle (global best) \mathbf{g}^* and a local leading particle (local best) \mathbf{b}_i^* . A particle i is defined by its position vector \mathbf{x}_i and velocity vector \mathbf{v}_i . Its new positions is updated by:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (1)$$

where

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \alpha \epsilon_1 \odot [\mathbf{g}^* - \mathbf{x}_i^t] + \beta \epsilon_2 \odot [\mathbf{b}_i^* - \mathbf{x}_i^t] \quad (2)$$

is the new velocity vector obtained with respect to \mathbf{g}^* , \mathbf{b}_i^* , and two random vectors ϵ_1 and ϵ_2 taking values between 0 and 1. The product of two matrices $\mathbf{u} \odot \mathbf{v}$ is a Hadamard entry-wise product defined as $[\mathbf{u} \odot \mathbf{v}]_{ij} = u_{ij}v_{ij}$. The constants α and β represent the learning factors or acceleration constants. They represent the attraction that a particle has either toward its own success or toward the success of its neighbours.

In order to accelerate the convergence of the algorithm, the velocity vector is generated based on global best by a simpler formula

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \alpha(\epsilon - \frac{1}{2}) + \beta(\mathbf{g}^* - \mathbf{x}_i^t) \quad (3)$$

where ϵ is a random vector with values from 0 to 1. Based on this change, update of the location of a particle is calculated without the recourse to a velocity vector by

$$\mathbf{x}_i^{t+1} = (1 - \beta)\mathbf{x}_i^t + \beta\mathbf{g}^* + \alpha(\epsilon - 0.5) \quad (4)$$

As explained in [14], this version will give the same order of convergence with typical values of $\alpha \approx 0.1 \sim 0.4$ and $\beta \approx 0.1 \sim 0.7$.

In applying PSO to the problem of optimal radar network, two important questions need to be addressed. The first question is how to deal with the vast space of solutions since in a finite geographical area we could end up with huge number of possible sites to choose from. The second question that need to be addressed is how to define an evaluation function which is easy and inexpressive to compute.

In our work, a modified explicit enumeration method is used similar to the one used by Kurdzo et al. [9]. The investigated geographical region is transformed into a gridded format with a resolution of approximately 0.09° (1km)

M latitudinal and N longitudinal spacing thus producing a matrix $\mathbf{A}_{M \times N}$. Each grid point is considered to be either a potential sitting site or a point covered by the radar signal. When a possible site is selected (at a latitudinal and longitudinal position), a theoretical circle is drawn representing the range indicated by the manufacturer of the radar. Once this range is established, the algorithm determines the grid points included in this theoretical circle. Using a binary encoding, all grid points are set to zero initially. The selected site plus the points included in the theoretical circle are all set to one. Hence if $a_{i,j}$ is a grid point where $i - j$ are latitude - longitude coordinates, then $a_{i,j} = 1$ if a point belongs to the circle and zero otherwise. The coverage area of a radar is the sum of all values of the grid points,

$$F = \sum_{i=1}^M \sum_{j=1}^N a_{i,j} \quad (5)$$

This fitness function would guarantee a much more spread out radars due to the fact that a point in the grid is illuminated only once even when covered by two or more radars. The boundary conditions can be defined either by a lower and upper latitude and longitude or by a state/ country border. A maximisation problem is then formulated using (5) as

$$\max \sum_{z=1}^R F_z \quad (6)$$

where R is the number of radars in the network.

The algorithm starts by initialising randomly P particles each representing a different combination of W radars placements. All P particles are evaluated using the fitness function (5) and the maximum P_{max} is selected using (6). If there is an improvement in the coverage area then the corresponding P_{max} particle is set as the global best and a new set of P particles are generated by (4). If the stopping condition is satisfied or the maximum number of iterations is reached then the algorithm is terminated.

3 Accounting for terrain blockage

When determining the theoretical coverage area of any radar, the manufacturer assumes the absence of any physical obstacles that could reduce or obstruct the propagation of a radar signal. This is not always the case as for example, placing a radar close to a high building will severely alter the propagation of the radar beyond that building. The topography of the geographical region is an important factor in coverage alterations of X-BAND radars. Introducing this factor to the illumination process of grid points covered by a signal radar enhances further the realism of radar coverage area.

Hence, to incorporate this new factor to our model, a terrain data source must be selected. United States Geological Survey provides a global digital elevation

data at a resolution of 30 arc seconds (≈ 1 kilometre) which is suitable for the utilised grid approach.

An assessment of signal propagation with respect to terrain blockage of each grid point representing a potential radar site or included inside the theoretical coverage area of a radar is performed using Zrnice and Doviak [3] 4/3 law:

$$h = \sqrt{r^2 + a_e^2 + 2ra_e \sin \theta_e} - a_e \quad (7)$$

where h is height of beam in km, r is range of beam in km, θ_e is the elevation angle, and a_e is the effective earth's radius in km ($4/3$ the earth's radius).

Starting from the grid point representing a potential radar site and propagating the radar beam in 1 km range gates, the illumination process of grid points is terminated if one of these conditions is met:

- the elevation at the actual grid point is greater than the height of the beam $h < \text{elevation}(i_{xy})$. fitness function
- the height of the beam exceeds a pre-set cutoff upper limit $h > \text{Upper limit}$
- the distance from the potential radar site is greater than the theoretical range limits set by the antenna manufacturer $r > \text{max range}$

The first condition implies that the radar beam has encountered an obstacle and can not propagate further beyond that point. The second condition allows the user to impose a limit on the height of radar beam. This means that if a user is interested only on weather system at altitude lower than 1 km, the upper limit is in this case set to 1 km. The last condition ensures that no grid point are illuminated beyond the theoretical range of the radar.

A significant change of the radar coverage area is observed when applying the three above conditions. The difference between the theoretical coverage area and the realistic area is nearly would range from 100% by placing a radar on position at an elevation higher than the pre-set maximum altitude to a 0% by installing the radar in perfectly flat region. By accounting for terrain blockage in the proposed algorithm, a more accurate representation of radar coverage is obtainable which in return leads to a more optimal placement of radars.

4 Numerical results

The analysis were conducted with a 1.1° radar beam elevation angle and a tower height of 15 m. A 1 km altitude upper limit to the radar beam is utilised while the theoretical range of the X-BAND radar is set to 45 km.

Two stopping criterion were imposed on the APSO algorithm, the first being pre-set maximum number of iterations while the second is a user selected value between 0 and 1 representing the desired percentage of area covered by the weather radar network.

For the results presented in Figure(1) 5 radars and a swarm of 5 particles were used. The selected geographical region is bounded by parallels 34°N and 36°N and meridians 3°E and 6°E with a total surface area of $6.076 \times 10^4 \text{ km}^2$. The

optimisation algorithm with 100 iterations was run 20 times producing different results ranging from a coverage percentage of $\approx 45\%$ to $\approx 52\%$ of the total area. Figure(1a) is a plot of the mean value of 20 fitness value computed at each iteration. With an average of $\approx 38\%$ as an initial start, the algorithm converges to an average of $\approx 50\%$ after only 100 iterations. Figure(1b) displays the optimal placement of the 5 radars with a realistic radar coverage of both the worst (in mauve) and best (in yellow) cases obtained from 20 different runs.

The second test consisted of placing optimally 50 WRs in the northern part of Algeria which is included inside the quadrangle bounded by parallels 32°N and 37.1°N and meridians 2°W and 9°E with a total area of $4.56 \times 10^5 \text{ km}^2$. Two different runs were performed, one with 10 particles and another with 25 particles. Figure(2a) displays the progress of the 500 iterations for the two runs. As expected, by increasing the swarm to 25 particles the method performed better from the initial iteration up to the final result by a deviation of $\approx 2\%$. The different optimal placements obtained from the two runs along with the analysed radar coverage area are presented in Figure(2b). The mauve polygons correspond to the result obtained with 10 particles while the yellow polygons are the outcome with 25 particles run.

The results displayed in Figure(3) were obtained using both a swarm with 5 particles and another with 15 particles to an optimal placement problem of 100 radars. The geographical area selected was the whole of Algeria with an area of more than $2.3 \times 10^6 \text{ km}^2$. A comparison of the fitness evaluation over 500 iterations for the two different cases show a slight increase in the coverage area when using 15 particles as seen in Figure(3a). Figure(3b) shows the optimal placement of the 100 WRs after 500 iteration for both 5 particles (in mauve) and 15 particles (in yellow).

As expected, the optimum placement of the radars are mostly located in flat lands rather than the mountainous region. Placing a radar in a site close or surrounded by mountains would restrict the propagation of its beam and thus limiting its coverage area.

Another important result is the reduction of the areas of intersection between different radars. All of Figures(1a,2b,3b) confirm a well spread out network of radars where intersection points are adequately minimal.

5 Conclusion

An APSO based algorithm for determining optimal placement of radars would constitute a perfect start for meteorologists to design a weather radar network. Given enough radar technical information and terrain data, the algorithm is capable of pinpointing suitable radar sitting sites at a relatively short time and with sufficient accuracy. This tool could reduce valuable time and cost through the reduction of suitable sites that are evaluated on field by experts.

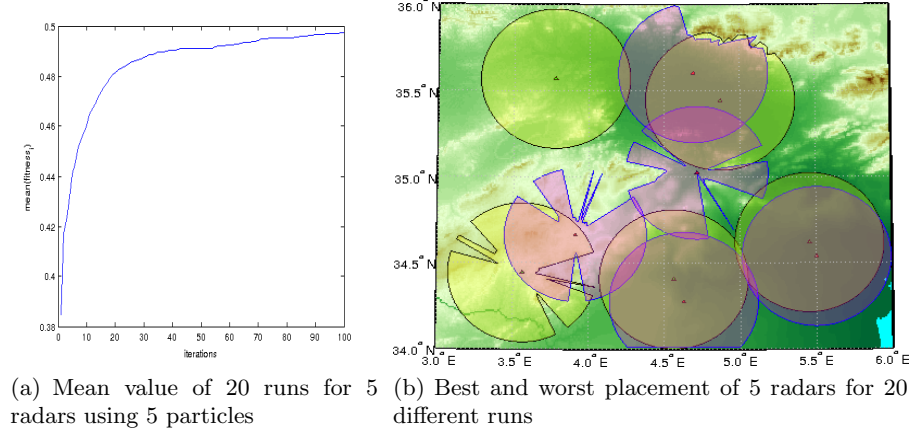


Fig. 1. Results of 20 different runs for 5 radars with 5 particles in the region LAT:[34°N 36°N] LON:[3°E 6°E]

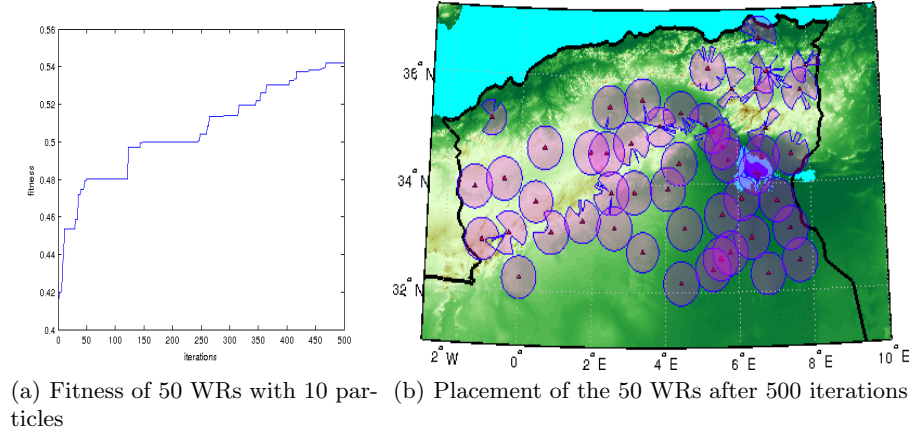


Fig. 2. Optimal placement of 50 WRs and a swarm of 10 particles after 500 iterations LAT:[32°N 37.1°N] LON:[2°W 10°E]

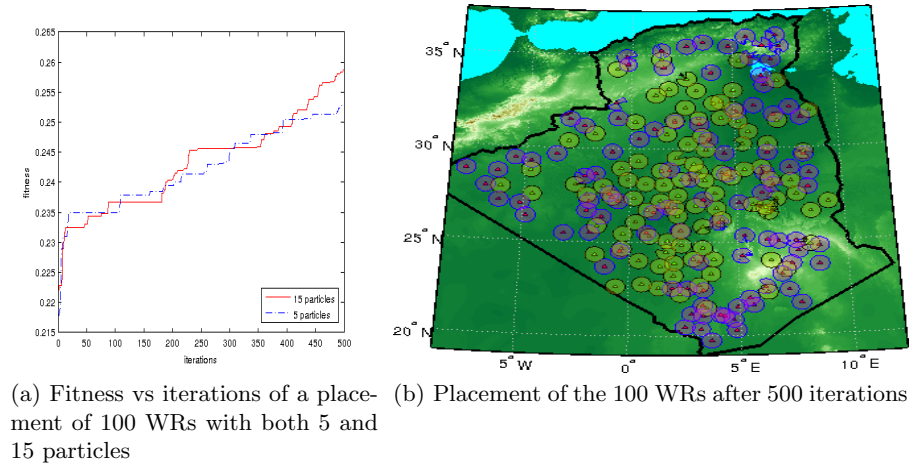


Fig. 3. Results obtained using the APSO approach for 100 WRs and a swarm of 5 and 15 particles after 500 iterations

References

1. B. A. Colle and C. F. Mass. Windstorms along the western side of the washington cascade mountains. part i: A high resolution observational and modeling study of the 12 february 1995 event. *Mountain Weather Review*, 126:28–51, 1998.
2. J. Petriceks R. T. H. Collis D. A. Leone, R. M. Endlich and J. R. Porter. Meteorological considerations used in planning the nexrad network. *Bulletin of American Meteorological Society*, 70:5–13, 1989.
3. R. J. Doviak and D. S. Zrnic. *Doppler Radar and Weather Observations*. Academic Press, 1984.
4. U. German and J. Joss. Spatial continuity of alpine precipitation. *Physical Chem Earth*, 25:903–908.
5. Iwan Holleman and Laurent Delobbe. The european weather radar network (opera). In *4th European Conference on Severe Storms*, September 2007.
6. H. Boynton J. Wilson, R. Carbone and R. Serafin. Operational application of meteorological doppler radar. *Bulletin of American Meteorological Society*, 61(0):1154 – 1168, 1980.
7. J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks, Piscataway, NJ*, page 1942, 1995.
8. J.C.M. Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional simplex space. *IEEE Transaction on Evolutionary Computation*, 6:5873, 2002.
9. James M. Kurdzo and Robert D. Palmer. On the use of genetic algorithms for optimization of a multi-band, multi-mission radar network. *Atmosphere*, 1(2), 2012.
10. K. L. Manross T. M. Smith P. L. Heinselman, D. L. Priegnitz and R. W. Adams. Rapid sampling of severe storms by the national weather radar testbed phased array radar. *Journal of Weather Forecasting*, 23:808–824, 2008.
11. Sittichoke Pookpant and Weerakorn Ongsakul. Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. *Journal of Renewable Energy*, 55:266–276, 2013.
12. Pyari Mohan Pradhan and Ganapati Panda. Connectivity constrained wireless sensor deployment using multiobjective evolutionary algorithms and fuzzy decision making. *Ad Hoc Networks*, 10:1134 – 1145, 2012.
13. R. Sacile R. Minciardi and F. Siccardi. Optimal planning of weather radar network. *Journal of Atmospheric and Oceanic Technology*, 20:1251 – 1262, 2003.
14. Xin-She Yang. *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley Publications, 2010.

BSG-Starcraft Radius improvements of Particle Swarm Optimization Algorithm: an application to Ceramic Matrix Composites

D. Chamoret¹, S. Salmon², N. Di Cesare¹ and Y.J. Xu³

¹ IRTES-M3M-EA7274, Université de Technologie de Belfort Montbéliard, Belfort, France

{dominique.chamoret,noelie.di-Cesare}@utbm.fr

² Optimization Command & Control Systems, Besançon, France

s.salmon@my-occs.fr

³ Engineering Simulation and Aerospace Computing, Key Laboratory of Contemporary Design & Integrated Manufacturing Technology, Northwestern Polytechnical University, Xi'an 710072, China

xu.yingjie@nwpu.edu.cn

Abstract. The thermal residual stresses (TRS) induced in ceramic matrix composites (CMCs) with multi-layered interphases when cooling down from the processing temperature, have a significant influence on the mechanical behavior and lifetime of CMCs. The objective of this work is to minimize the TRS of the unidirectional CMCs with multi-layered interphases by controlling the interphases thicknesses. A new Particle Swarm Optimization (PSO) algorithm is interfaced with a finite element code to find an optimal design and thereby significantly reduce the TRS within CMCs. This new PSO allows a faster convergence rate and gets a new effective stopping criteria based on real physical limits.

Keywords: Ceramic matrix composites, Thermal residual stresses, Particle Swarm Optimization, Radius improvement, the BSG-Starcraft improvement, Microstructure modelling, Finite element analysis

1 Introduction

The Particle Swarm Optimization (PSO) algorithm belongs to the category of swarm intelligence techniques. In PSO, each solution of the optimization problem is regarded as a particle in the search space, which adjusts its position in the search space according to its own flying experience and the flying experience of other particles [1]. The PSO algorithm has only a small number of parameters which need to be adjusted and is easy to implement. However, this algorithm has a major drawback: the number of iterations needed to find a potentially global minimum. In practical situations such as the structural optimization context, the optimization techniques may be linked to the finite element method. For this type of problems, the evaluation of the cost function for given values of

the design variables requires a finite element analysis. This work can be very CPU time consuming especially when the finite element models are large and have a considerable number of design parameters. In this paper, a new criterion is developed. These improvements, BSG-Starcraft Radius improvements, can drastically reduce the CPU time by avoiding needless iterations.

To test our approach, an original application to composites (CMCs, Ceramic matrix composites) is proposed. Ceramic matrix composites (CMCs) with multi-layered interphases exhibit attractive properties for thermal-structural applications [2, 3]. However, in CMCs with multi-layered interphases, thermal residual stresses (TRS) are often generated upon cooling from processing to room temperatures due to extensive mismatch of the coefficients of thermal expansion between the constituents (fiber, interphase and matrix). The distribution of TRS, resulting in the cracks and separations in the matrix and interphases, has a significant influence on the mechanical behavior and lifetime of CMCs. The aim of the presented example is the optimization by PSO of TRS distribution in the multi-layered (PyC/SiC)_n interphases and matrix from the point of view of the deposition thickness of each interphase layer [4] in order to achieve an excellent thermal-mechanical performance of CMCs.

2 Particle Swarm Optimization (PSO)

The PSO algorithm is a global optimization algorithm described as sociologically inspired. In PSO, each individual of the swarm is considered as a particle in a multi-dimensional space that has a position and a velocity. These particles fly through hyperspace and remember the best position that they have seen.

2.1 Basis of particle swarm algorithm

Members of a swarm fly in the search field (of N dimensions) and each member is attracted by its personal best solution and by the best solution of its neighbours [5, 6]. Each particle has a memory storing all data relating to its flight (location, speed and its personal best solution). It can also inform its neighbours, i.e. communicate its speed and position. This ability is known as socialisation. For each iteration, the objective function is evaluated for every member of the swarm. Then the leader of the whole swarm can be determined: it is the particle with the best personal solution. The process leads at the end to the best global solution.

At each iteration t , the location and speed of one particle x_i are updated as follows [1]:

$$\begin{cases} v_i^{t+1} = \omega v_i^t + r_1 c_1 (p_g^t - x_i^t) + r_2 c_2 (p_i^t - x_i^t) \\ x_i^{t+1} = v_i^{t+1} + x_i^t \end{cases} \quad (1)$$

where v_i and x_i represent the current velocity and the position of the i th particle respectively (note that the subscripts t and $t+1$ refer to the recent and the next iterations respectively). p_i is the personal best previous position (*pbest*) of the i th particle and p_g is the best global position (*gbest*) among all the particles

in the swarm. The parameters r_1 and r_2 are two random numbers between 0 and 1. The constant c_1 and c_2 represent trust parameters indicating how much confidence the current particle has in itself and how much confidence it has in the swarm. These acceleration constants c_1 and c_2 indicate the stochastic acceleration terms which pull each particle toward the best position attained by the particle or the best position attained by the swarm. Low values of c_1 and c_2 allow the particles to wander far away from the optimum regions before being tugged back, while the high values pull the particles toward the optimum or make the particles to pass through the optimum abruptly. In reference [7], the constants c_1 and c_2 are chosen equal to 2 corresponding to the optimal value for the problem studied. The role of the inertia weight ω is considered important for the convergence behaviour of PSO algorithm. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity. Thus, the parameter ω regulates the trade off between the global (wide ranging) and the local (nearby) exploration abilities of the swarm. A proper value for the inertia weight provides balance between the global and local exploration ability of the swarm, and thus results in better solutions. Numerical tests imply that it is preferable to initially set the inertia to a large value, to promote global exploration of the search space, and gradually decrease it to obtain refined solutions [8].

2.2 PSO improvements

The two improvements of the Particle Swarm Optimization algorithm presented here have been developed by the author S. Salmon in [9] and implemented in Scilab [10]. In this paper, only the algorithms are presented. The efficiency of the proposed methods has been evaluated and the methods have been tested statistically compared to the classical PSO using usual benchmarks in [10]. Moreover, recently, the new version of PSO was tested and validated in a few original cases [11].

The Radius improvement This improvement is based on practical considerations: the optimization process is stopped when the swarm became too small to show any influences on the behavior of a real system. In our example, a structural optimization problem, a thickness precision value under $0.1\mu m$ is not possible due to multiple factors such as milling precision. So it became useless to continue to optimize our model, as it will be impossible to manufacture the optimized solution. In a general way, a minimum radius is defined for each design variable and while the swarm radius (using the L_∞ norm) is higher than the radius limits, the optimization process goes on. When all radius values are under our defined limits, a counter starts for 10 iterations. During this time, if all radius are kept under limits the optimization process stops and save computation time. Otherwise, the counter is reset (Figure 1(a)).

The BSG-Starcraft improvement The Battlestar Galactica Starcraft (BSG-Starcraft) improvement is based on two ideas inspired from the science fiction

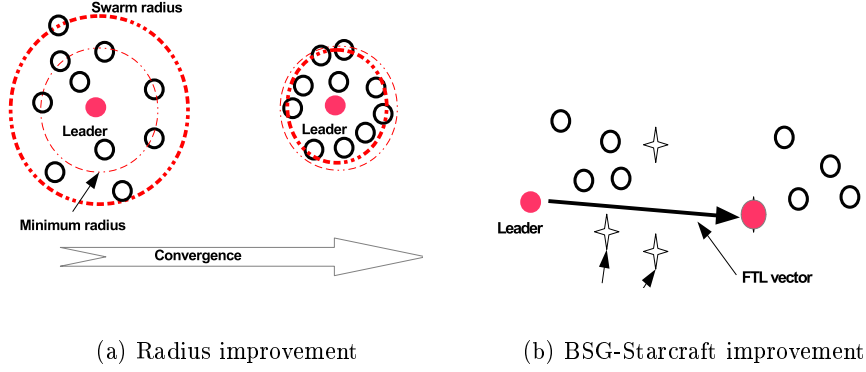


Fig. 1. PSO improvements

film Battlestar Galactica and a video game Starcraft. These two ideas can be formulated as follows:

- at each iteration t , the particle leader g has the possibility to send randomly some new particles to fast explore the space, the raptors (Figure 1(b)).
- if one raptor finds a best position than the global best then the swarm jumps (FTL jump), conserving the swarm geometry, to this new location. The carrier location is now the raptor one.

More details of the approach are given by the pseudo-code listed in algorithm 1. In this paper, number of raptors are set equal to the particle number of the swarm.

3 Application to Ceramic Matrix Composites

3.1 Context

The fabrication process of unidirectional CMCs with multi-layered (PyC/SiC) interphases is briefly introduced below: the architectures of CMCs consist of arranged fibers. The components of the multi-layered (PyC/SiC) n interphases and the SiC matrix are infiltrated within the porous fiber preforms. In the present study, square fiber arrays are used to model the unidirectional CMCs. Four layers of interphases are distributed around the fibers. In the longitudinal direction, the fiber axes have been assumed to be parallel and of equal lengths (Figure 2). The optimization study is carried out on an unidirectional SiC fiber reinforced SiC ceramic matrix composite with four alternate layers of (PyC/SiC/PyC/SiC) interphases. Figure 2 shows the unit cell model of the composite. ϕ_f is fiber diameter; $d_1 \sim d_4$ are thicknesses of the interphase layers; d_5 is the thickness of the matrix layer. The finite element method is used for numerical computation of TRS. The finite element model can be seen in figure 2. Material properties of the constituents are given in Table 3.1.

Algorithm 1 Particle swarm optimization : BSG-Starcraft improvement**Require:** Initialization

- 1: Initialize population : randomly initialize positions of all particles distributed throughout the design space.
- 2: Initialize weight of all particles
- 3: Initialize velocities of all particles
- 4: Evaluate the objective function for all particles
- 5: Find the personal best : $p_i^0 = x_i^0$
- 6: Find the global best: determine $\min f(x_i^0), i = 1 \cdots n$, set $p_g^0 = x_{min}^0$
- 7: **while** $t \leq t_{max}$ **do**
- 8: **for** $i = 1, n$ **do**
- 9: Find the global best of the previous iteration : it is the carrier $x_{carrier}$
- 10: Create randomly, for a quite long range from the carrier, 20 raptors to explore the space x_{raptor}
- 11: Evaluate the objective function for the raptors $f(x_{raptor})$
- 12: **if** $f(x_{raptor}) \leq f(x_{carrier})$ **then**
- 13: Jump the swarm conserving geometry
- 14: Evaluate the objective function for the jumped swarm
- 15: **else**
- 16: Evaluate the objective function of the initial swarm
- 17: **end if**
- 18: Update the personal best
- 19: Update the global best
- 20: Update velocity
- 21: Update position
- 22: **end for**
- 23: **end while**

3.2 Optimization problem

The goal of this work is the optimization of TRS distribution in the multi-layered interphases and matrix from the viewpoints of the deposition thickness of each interphase layer. Mathematically, the optimization problem can be formulated as:

$$\begin{cases} \min & TRS = f(X) \\ X = & (d_1, d_2, d_3, d_4) \\ 0.3 \leq & d_i \leq 0.6 \end{cases} \quad (2)$$

where $f(X)$ is the objective function, i.e. the maximum hoop TRS within the interphases and matrix. The vector X is the vector defining the design variables: the thicknesses of the interfaces.

The diameter of the SiC fibre is $10\mu m$ and the thickness of the SiC matrix is $2\mu m$. The upper bound of each interphase layer thickness is $0.6\mu m$. In practice, the thicknesses of multi-layered interphases are usually limited to $0.1\mu m$ or more for oxidation resistance considerations [12] and reduction of the complexity of the CVI fabrication process. Therefore, in the present study the lower bound for each interphase layer thickness is set to $0.3\mu m$.

Constituent	E_{11} (GPa)	E_{33} (GPa)	G_{12} (GPa)	G_{23} (GPa)	ν_{12}	ν_{23}	α_{11} ($10^{-6}/^{\circ}\text{C}$)	α_{33} ($10^{-6}/^{\circ}\text{C}$)
SiC fibre	200	200	80	80	0.12	0.12	3	3
PyC interphase	12	30	4.3	2	0.4	0.12	28	2
SiC interphase	350	350	145.8	145.8	0.2	0.2	4.6	4.6
SiC matrix	350	350	145.8	145.8	0.2	0.2	4.6	4.6

Table 1. Properties of the constituents

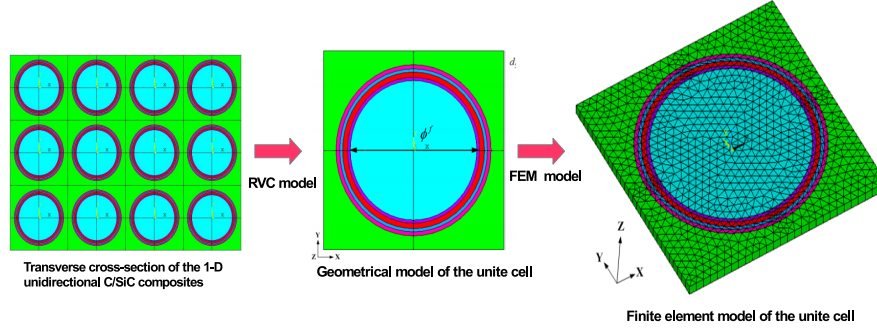


Fig. 2. Geometrical model of the unit cell of SiC/SiC composite

In this problem, the evaluation of the objective function for given values of the design variables requires a finite element analysis. So, the PSO and BSG-Starcraft Radius PSO schemes are linked to the finite element model introduced before (Figure 3).

3.3 Results

For all these algorithms, a population of 20 individuals is used; the inertia weight ω decreases linearly from 0.9 to 0.4. The value of constants c_1 and c_2 are set to be the same and equal to 2. The maximum number of iterations is limited to 200. Radius value is set to $0.1\mu\text{m}$.

The results obtained by our approach are compared with those obtained by a classical PSO in order to evaluate the performance of the new algorithm. Figure 4 provides a convergence rate of the optimization procedure. It can be seen that both two algorithms achieve the best solutions. The maximum hoop TRS has been decreased to 0.22GPa by means of handling the interphases thicknesses. The final optimized interphases thicknesses are $(0.6\mu\text{m}, 0.6\mu\text{m}, 0.3\mu\text{m}, 0.6\mu\text{m})$. However, the BSG-Starcraft Radius PSO algorithm displays a faster convergence rate than the PSO algorithm in this example. It is closer to the best solution than the PSO algorithm in the early stages and hence. Besides, due to the swarm radius limitation the BSG-Starcraft Radius PSO algorithm stopped the

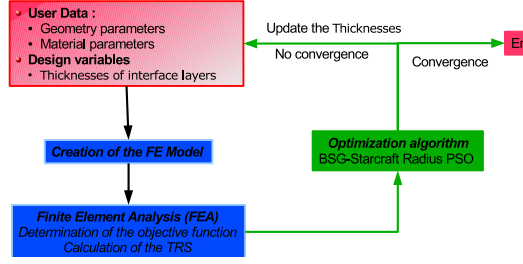


Fig. 3. Optimization Process

computation after 97 iterations. Hence, the BSG-Starcraft Radius PSO algorithm requires less computational effort to find best design solutions than the PSO algorithm.

To conclude the comparison between the PSO and BSG-Starcraft Radius PSO schemes, the CPU time has been evaluated. An optimization run with the classical PSO algorithm takes approximately 12.5 hours on 7 CPU cores while the same run with the improvements takes 8.08 hours. A drastic decrease of the CPU time can be seen when the BSG-Starcraft Radius PSO scheme is used. In a structural optimization context where the optimization algorithm is coupled to a finite element analysis, the observed reduction of the number of iterations affects the number of finite element analysis and so the CPU time is strongly reduced.

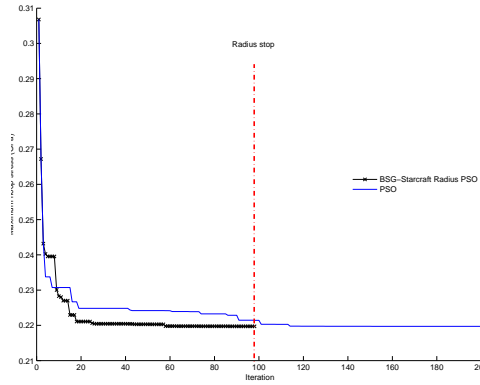


Fig. 4. Convergence rate for the optimization of hoop TRS

4 Conclusion

PSO algorithm does not use the gradient information of the optimization problem. This means that it does not require for the optimization problem to be differentiable as required by classical optimization methods such as gradient descent or quasi-Newton methods. In this structural optimization context, the optimization algorithm is coupled with a finite element analysis to evaluate the cost function. This work can be very CPU time consuming especially when the finite element models are large and have a considerable number of design parameters. That is the reason why the new version of PSO proposed in this paper is very interesting by its faster convergence rate and its stopping criteria based on real physical limits. An original application to composite is proposed to test and validate our approach.

References

1. Kennedy, J., Eberhart, R.: Particle swarm optimisation. In: Proceedings of the IEEE International Conference on Neural Networks. (1995)
2. Droillard, C., Lamom, J., Bourrat, X.: Strong interface in cmcs, a condition for efficient multilayered interphases. In: Materials Research Society Symposium Proceedings. Volume 365. (1995) 371–376
3. Droillard, C., Lamon, J.: Fracture toughness of 2-d woven sic/sic cvi-composites with multilayered interphases. *Journal of the American Ceramic Society* **79**(4) (1996) 849–858
4. Xu, Y., Zhang, W., Chamoret, D., Domaszewski, M.: Minimizing thermal residual stresses in c/sic functionally graded material coating of c/c composites by using particle swarm optimization algorithm. *Computational Materials Science* **61** (2012) 99 – 105
5. Janson, S., Middendorf, M.: On trajectories of particles in pso. In: Proceedings of the 2007 IEEE Swarm Intelligence Symposium. (2007)
6. van den Bergh, F., Engelbrecht, A.: A study of particle swarm optimization particle trajectories. *Information Sciences* **176**(8) (2006) 937 – 971
7. Cristian, I., Trelea: The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters* **85**(6) (2003) 317 – 325
8. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on. (1998) 69–73
9. Salmon, S.: Caractérisation, identification et optimisation des systèmes mécaniques complexes par mise en œuvre de simulateurs hybrides matériels-logiciels. PhD thesis, Université de Technologie de Belfort Montbéliard (2012)
10. Salmon, S.: Particle Swarm Optimization in Scilab ver 0.1-7 – Performance evaluations. (2011)
11. Göktürkler, G., Balkaya, .: Inversion of self-potential anomalies caused by simple-geometry bodies using global optimization algorithms. *Journal of Geophysics and Engineering* **9**(5) (2012) 498
12. Bertrand, S., Pailler, R., Bourrat, X., Naslaj, R.: Tem structure of (pyc/sic)_n multilayered interphases in sic/sic composites. *Journal of the European Ceramic Society* **20** (2000) 1–13

Combining PSO and FCM for Dynamic Fuzzy Clustering Problems

Yucheng Kao*, Ming-Hsien Chen and Kai-Ming Hsieh

Department of Information Management, Tatung University
No. 40, Sec. 3, Zhongshan N. Rd., Taipei, Taiwan
*ykao@ttu.edu.tw

Abstract. This paper proposes a dynamic data clustering algorithm, called PSOFC, in which Particle Swarm Optimization (PSO) is combined with the fuzzy c-means (FCM) clustering method to find the number of clusters and cluster centers concurrently. Fuzzy c-means can be applied to data clustering problems but the number of clusters must be given in advance. This paper tries to overcome this shortcoming. In the evolutionary process of PSOFC, a discrete PSO is used to search for the best number of clusters. With a specified number of cluster, each particle employs FCM to refine cluster centers for data clustering. Thus PSOFC can automatically determine the best number of clusters during the data clustering process. Six datasets were used to evaluate the proposed algorithm. Experimental results demonstrated that PSOFC is an effective algorithm for solving dynamic fuzzy clustering problems.

Keywords: Particle Swarm Optimization, Fuzzy c-means, Data Clustering

1 Introduction

Data clustering aims to discover meaningful organization of data points in a data set in terms of their similarities and dissimilarities. A good clustering algorithm can classify a set of data points into several distinct clusters such that the members of a cluster are highly similar while the data points belonging to different clusters are dissimilar. Data clustering is an important data mining technique and has been studied in several fields such as pattern recognition, machine learning, market segmentation, bioinformatics, and so on.

Data clustering approaches can be roughly classified into two main categories: hierarchical clustering and partitional (non-hierarchical) clustering [3]. Partitional clustering methods require the number of clusters to be given beforehand and use an iterative algorithm to find out the best cluster centers for classifying data points into appropriate clusters. K-means and fuzzy c-means [2] are well-known partitional clustering methods. K-means is for crisp clustering while FCM is for fuzzy clustering. When using a partitional clustering approach, we have to provide the number of clusters and initial cluster centers in advance. Initial cluster centers can be randomly determined, but it is difficult for us to

decide the best number of clusters in priori unless we understand the characteristics of data sets very well. To overcome the drawback of partitional clustering, several papers have proposed different approaches, such as genetic algorithm (GA)-based approach [1] and PSO-based approach [8].

This paper proposes a PSO-based fuzzy clustering (PSOFC) approach to deal with dynamic clustering problems. When running PSOFC, we only need to give a maximum possible cluster number in priori. In the evolutionary process, each particle selects its own best number of clusters and refines the corresponding cluster centers using fuzzy c-means. We use a clustering validity index to evaluate the clustering results of particles, which in turn are used to direct the search directions of particles. The experimental results show that PSOFC can effectively find the best clustering results, compared with current dynamic clustering algorithms.

2 Background

In this section, we introduce the PSO algorithm, fuzzy c-means, and a cluster validity measure (PBM Index).

2.1 Particle Swarm Optimization

PSO was originally proposed by Kennedy and Eberhart in 1995 [5]. The concept of PSO follows a biological swarm behavior model which was inspired by the behavior of birds searching for food sources. When birds are foraging for food, they memorize past best positions and exchange the message of best positions with each other so that all members can gradually fly toward the best target. In a PSO algorithm, a candidate solution is called a particle. Each particle memorizes two components: the personal best (pBest) and global best (gBest) positions. The former is the best position that a particle has ever visited while the latter is the best position that the whole swarm has ever experienced. A particle determines its new velocity and position according to its current velocity, its pBest position and the gBest position, as defined in Eqs. (1) and (2).

$$V_{id}^{t+1} = w^{t+1} \times V_{id}^t + c_1 \times rand() \times (P_{id}^t - X_{id}^t) + c_2 \times rand() \times (P_g^t - X_{id}^t) \quad (1)$$

$$X_{id}^{t+1} = X_{id}^t + V_{id}^{t+1} \quad (2)$$

where V_i is the velocity of particle i , X_i is the position of particle i , P_i is pBest of particle i , P_g is gBest of the particle swarm, c_1 and c_2 are positive constants, $rand()$ is a random number selected from $[0.0, 1.0]$, d is the dimension index, and t is the generation number.

The advantages of using PSO include fast convergence and fewer parameters needed to be set. Some researchers have applied PSO to data clustering [6] [7] [8].

2.2 Fuzzy c-Means clustering algorithm

Fuzzy c-means is a clustering method similar to K-means but the concept of fuzzy theory is incorporated to improve clustering results. That is, fuzzy c-means allows that each data point belongs to more than one cluster according to their fuzzy memberships. Assume we are going to classify n data objects into c groups, the objective function used in FCM is defined below:

$$J_m = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m [dist(c_i, x_j)]^2 \quad (3)$$

where c_i is the center vector of cluster i , x_j is data point j , u_{ij} is membership degree of point x_j related to cluster i , m is the fuzziness index and its value falls in the range of $[1, \infty]$, $dist(c_i, x_j)$ is Euclidean distance between data point j and cluster center i .

The FCM algorithm contains following main steps:

Step 1: Initialize the parameters. Note that the values of u_{ij} are subject to the following three conditions. Condition (a) means that the value of u_{ij} is a real number ranged between 0 and 1. Condition (b) indicates that for a data point, all of its u_{ij} values should be sum up to 1. The last condition defines that the sum of u_{ij} values for a cluster should be ranged between 0 and n .

(a) $u_{ij} \in [0, 1] \quad i = 1, 2, \dots, c \text{ and } j = 1, 2, \dots, n$

(b) $\sum_{i=1}^c u_{ij} = 1 \quad j = 1, \dots, n$

(c) $0 < \sum_{j=1}^n u_{ij} < n \quad i = 1, \dots, c$

Step2: Calculate the center vector of cluster i by Eq. (4).

$$c_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m} \quad i = 1, \dots, c \quad (4)$$

Step3: Update the fuzzy membership matrix U by Eq. (5).

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{dist(c_i, x_j)}{dist(c_k, x_j)} \right)^{2/(m-1)}} \quad i = 1, \dots, c \text{ and } j = 1, \dots, n \quad (5)$$

Step4: Calculate J_m using Eq. (3) and check the stop criteria. If $|J_m^{t+1} - J_m^t| < \varepsilon$, then stop the execution; otherwise go back to step 2.

2.3 Cluster Validity Measure

When solving a dynamic clustering problem, a validity index is used to determine the best clustering solution. A good validity measure considers the degrees of compactness and separation in order to ensure that the optimal number of clusters is found. This paper uses PBM Index as the cluster validity measure [9]. PBM formulas are defined as follows:

$$PBM(K) = \left(\frac{1}{K} \times \frac{E_1}{E_K} \times D_k \right)^2 \quad (6)$$

$$E_1 = \sum_{j=1}^n \text{dist}(c_0, x_j) \quad (7)$$

$$E_K = \sum_{k=1}^K E_k = \sum_{k=1}^K \sum_{j=1}^n u_{kj}^m \text{dist}(c_k, x_j) \quad (8)$$

$$D_k = \max(\text{dist}(c_p, c_q)) \quad p, q = 1, \dots, k \quad (9)$$

where K is the total number of clusters, E_1 is the sum of the distances between each data point and the geometric center of the data set (c_0), x_j is data point j , c_0 is the cluster center when the number of clusters is one, E_K is the sum of within-cluster distances of K clusters, and D_k is the maximum separation distance of each pair of cluster centers. Eq. (6) suggests that the higher the PBM Index value, the better the clustering result.

3 PSOFC Clustering Algorithm

In PSOFC, we adopt a discrete PSO algorithm to search for a best number of clusters and utilize the fuzzy c-means to find optimal cluster centers for data clustering. The clustering result is evaluated by computing cluster validity index PBM.

3.1 Solution Representation

PSOFC considers each particle as a candidate solution, which is composed of the number of clusters and several cluster centers. The number of clusters is not given in advance and must be determined at each iteration. Thus the lengths of particle solutions are varied. The number of clusters should be within a reasonable range, depending on the size of the data set. The minimum cluster number ($MinK$) is set to 2, and the maximum cluster number ($MaxK$) is set to \sqrt{n} [11]. The solution representation of particle i with $K_i = k$ is defined below:

$$P_i = [K_i = k; (x_{i11}, x_{i12}, \dots, x_{i1d}), (x_{i21}, x_{i22}, \dots, x_{i2d}), \dots, (x_{ik1}, x_{ik2}, \dots, x_{ikd})]$$

where x_{i11} is the first dimension of the first cluster center and d is the dimension size of data points.

3.2 Initialization

First, each particle randomly determines its own cluster number K_i and initial flying velocity V . Here, the value of K_i is selected from the interval of $MinK$ and $MaxK$, i.e. $MinK \leq K_i \leq MaxK$. The initial velocity is randomly selected from the interval of -1 and 1 . According to the specified cluster number, each particle randomly selected K_i data points from the data set as its initial cluster

vectors. After that, each particle executes the fuzzy c-means clustering algorithm to refine cluster centers, and at last uses the Eq. (6) to calculate PBM value. According to the PBM values, the best particle is determined and its cluster number and cluster centers are the swarm best solution, i.e. gBest. Each particle also memorizes its current solution as pBest.

3.3 Main Steps of the Evolution Process

There are four main steps in an iteration of PSOFC, which aims to find the best number of clusters and the corresponding cluster centers.

Step 1: Evolving K_i values As we know, determining the best number of clusters for a clustering problem is a discrete optimization problem. Therefore, a discrete PSO algorithm proposed by Jarboui et al. [4] is adopted to determine the cluster number for each particle. The main concept is to convert the number of clusters of particle i (K_i) to a virtual space using a state variable (Y_i) (see Eq. (10)), to use a conventional PSO algorithm to compute the new velocity and position of state variable Y_i in the virtual space (see Eqs. (11) and (12)), and then to convert the state variable with a new value back to the discrete space to get new K_i (see Eqs. (13) and (14)).

$$Y_i^t = \begin{cases} 1 & \text{if } K_i^t = P_g^t, \\ -1 & \text{if } K_i^t = P_i^t, \\ -1 \text{ or } 1 & \text{if } K_i^t = P_g^t = P_i^t, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

$$V_i^{t+1} = w^{t+1} \times V_i^t + c_1 \times rand() \times (-1 - Y_i^t) + c_2 \times rand() \times (1 - Y_i^t) \quad (11)$$

$$\lambda_i^{t+1} = Y_i^t + V_i^{t+1} \quad (12)$$

$$Y_i^{t+1} = \begin{cases} 1 & \text{if } \lambda_i^{t+1} > \alpha^{t+1}, \\ -1 & \text{if } \lambda_i^{t+1} < -\alpha^{t+1}, \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

$$K_i^{t+1} = \begin{cases} P_g^t & \text{if } Y_i^{t+1} = 1, \\ P_i^t & \text{if } Y_i^{t+1} = -1, \\ MinK \leq RN \leq MaxK & \text{otherwise.} \end{cases} \quad (14)$$

where P_g^t is the past best cluster number of the particle swarm, P_i^t is the past best cluster number of particle i , Y_i^t is the state variable of particle i , λ_i^t is the continuous version of Y_i^t , α is a threshold value, RN is an integer random number selected from the interval of $(MinK, MaxK)$.

- Step 2: Selecting new cluster centers** It could happen to FCM to fall into a local optimum if initial cluster centers are not well chosen. To avoid this drawback, PSOCF changes cluster centers in different ways, depending on the value of state variable Y . If $Y_i^{t+1} = 1$, particle i will replace its cluster centers with those of the gBest particle. In addition, some cluster centers will be randomly selected and substituted with data points randomly selected from the input data set. If $Y_i^{t+1} = -1$, particle i will replace its cluster centers with those of its pBest solution. The same, several cluster centers will be randomly selected and replaced with data points randomly selected from the input data set. If $Y_i^{t+1} = 0$, particle i will replace its all cluster centers with K_i^{t+1} data points randomly selected from the input data set.
- Step 3: Refining cluster centers with FCM** When new cluster centers of particle i are determined, PSOCF starts to refine the cluster centers using FCM (see Eqs. (3), (4) and (5)).
- Step 4: Updating pBest and gBest** After performing FCM, PSOCF calculates PBM Index value for each particle solution. Then the algorithm updates the personal best solution for each particle and the global best solution for the whole population according to their PBM Index values. At the end of an iteration, termination condition is checked. If the maximum iteration number is reached, then stop running the algorithm; otherwise, go back to step 1 to continue next iteration.

Table 1. Dataset Information.

Dataset	Number of Attributes	Number of points	Actual number of clusters
Data_4_3	3	400	4
Data_4_2	2	80	4
Data_5_2	2	250	5
Date_6_2	2	300	6
Iris	4	150	3
Breast Cancer	9	683	2

4 Experiments

In order to validate the clustering performance of the PSOCF algorithm, four artificial data sets and two real world data sets from UCI machine learning repository [10] were selected as test problems. The detailed information of these six data sets is listed in Table 1.

The PSOCF algorithm was coded in Visual C#, and all experiments were run on a personal computer with Pentium IV(3.0 GHz) running Windows XP. Based on the results of preliminary experiments, the parameters of PSOCF were set as follows: $MinK = 2$, $MaxK = \sqrt{n}$ (n is the number of data points in a test data set), the population size of particles $S = 20$, the stop criteria of fuzzy c-means $\varepsilon = 0.001$, $m = 1.5$, $\alpha = 1.0 \sim 0.35$, $c_1 = c_2 = 0.5$, $w = 0.72 \sim 0.4$,

the maximum number of iterations $MNI = 30$. Here $\alpha = 1.0 \sim 0.35$ denotes that the parameter will be linearly decreased from 1.0 to 0.35 during a run. To compare PSOFC with current algorithms, two dynamic clustering algorithms were selected from the literature. The first one is GCUK [1] and the second one is DCPSO [8]. The parameters of GCUK were set as follows: the number of chromosomes is 50, the number of iterations is 100, crossover rate is 0.8, mutation rate is 0.001, $MinK$ is 2, and $MaxK$ is \sqrt{n} . On the other hand, the parameters of DCPSO were set as follows: $N_c = 20$, $P_{ini} = 0.75$, $s = 100$, $w = 0.72$, $c_1 = c_2 = 1.49$, $V_{max} = 255$, MNI of binary PSO = 50 and the number of outer iterations is 2. For each of the six test data sets, 20 independent runs were performed for each of these algorithms.

The experimental results are summarized in Table 2. The results show that the performance of PSOFC is better than that of two existing algorithms in terms of the average values of objective function. Regarding the number of clusters found by these algorithms, it is obvious that PSOFC is able to provide the best number of clusters compared with GCUK and DCPSO. For example, the real-world data set Iris has two overlapped data clusters. PSOFC can correctly identify three clusters in this test problem while GCUK and DCPSO cannot.

Table 2. Experimental results.

Dataset	Algorithm	Objective		Obtained Cluster number	Actual cluster number	CPU Time	
		Function					
		Value					
		Avg.	Stdev			Avg.	Stdev
Data_4.2	GCUK	11.7365	3.1135	2,3	4	1.3761	0.1664
	DCPSO	13.2984	1.2046	2		2.5102	0.1283
	PSOFC	18.4572	0.0732	4		1.1477	0.0139
Data_5.2	GCUK	21.6571	3.5126	3,5	5	4.7211	0.4276
	DCPSO	26.588	0.0571	5		5.0953	0.3122
	PSOFC	26.8756	0.0658	5		3.7773	0.0328
Data_6.2	GCUK	372.341	4.3122	3,4,5	6	5.4122	0.3017
	DCPSO	408.7406	2.0281	5		5.932	0.2043
	PSOFC	557.411	0.0483	6		3.6828	0.0418
Data_4.3	GCUK	651.7921	3.7683	2,3,4	4	5.2189	0.4261
	DCPSO	731.9498	0.3181	4		5.9922	0.2007
	PSOFC	732.0347	0.0374	4		3.7336	0.1449
Iris	GCUK	20.8765	1.3712	2	3	3.9118	0.2132
	DCPSO	21.3497	1.2215	2		4.2117	0.1943
	PSOFC	27.1652	0.0451	3		3.5664	0.0587
Breast Cancer	GCUK	169.6934	0.8976	2	2	15.5612	0.5164
	DCPSO	169.2878	0.9735	2		17.7039	0.4952
	PSOFC	170.3247	0.0327	2		10.4805	0.5321

5 Conclusion

This paper proposed a new fuzzy clustering algorithm combining PSO and FCM to deal with dynamic clustering problems. By sharing the best results among particles, the optimal number of clusters and cluster centers are easily found. From the experimental results, it can be seen that PSOFC can find the best cluster number even though the number of clusters is not given in advance. The results also demonstrate that PSOFC has the ability to provide equal or better solutions for the test problems. For the future work, it is worth to apply other validity measure indices to the same clustering problems to see if the clustering results are sensitive to different validity measure indices.

Reference

1. Bandyopadhyay, S., Maulik, U.: Genetic clustering for automatic evolution of clusters and application to image classification. *Pattern Recognition* 35(6), 1197–1208 (2002)
2. Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, NY (1981)
3. Han, J., Kamber, M., Pei, J.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3 edn. (2011)
4. Jarbouli, B., Cheikh, M., Siarry, P., Rebai, A.: Combinatorial particle swarm optimization (cpso) for partitional clustering problem. *Applied Mathematics and Computation* 192(2), 337–345 (2007)
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*. vol. 4, pp. 1942–1948. Perth, Australia (1995)
6. Van der Merwe, D.W., Engelbrecht, A.P.: Data clustering using particle swarm optimization. In: *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*. vol. 1, pp. 215–220. IEEE (2003)
7. Mir, M.G., Tadayon Tabrizi, G.: Improving data clustering using fuzzy logic and pso algorithm. In: *Electrical Engineering (ICEE), 2012 20th Iranian Conference on*. pp. 784–788. IEEE (2012)
8. Omran, M.G.H., Engelbrecht, A.P., Salman, A.: Dynamic clustering using particle swarm optimization with application in unsupervised image classification. *Transaction on Engineering, Computing and Technology* 9, 199–204 (2005)
9. Pakhira, M.K., Bandyopadhyay, S., Maulik, U.: Validity index for crisp and fuzzy clusters. *Pattern recognition* 37(3), 487–501 (2004)
10. UCI: UCI Repository of Machine Learning Databases, <http://archive.ics.uci.edu/ml/>
11. Yang, S.L., Li, Y.S., Hu, X.X., Pan, R.Y.: Optimization study on k value of k-means algorithm. *Systems Engineering-Theory & Practice* 26(2), 97–101 (2006)

Hybrid ACO-SA algorithm for the Multi-objective Frequency Assignment Problem in Broadcasting

A. Bedoui^{1,2,*}

¹ TDF, 1 rue Marconi, F-57078 Metz, France

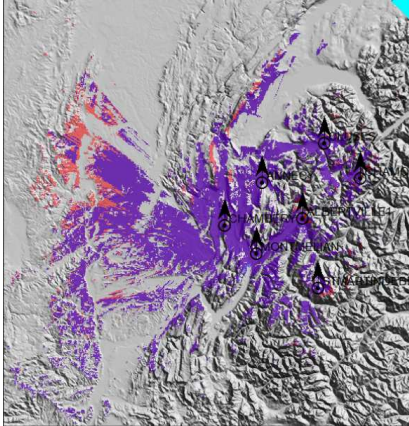
² LORIA, Campus V. Grignard, F-54506 Vandœuvre-lès-Nancy, France
akram.bedoui@tdf.fr

ABSTRACT

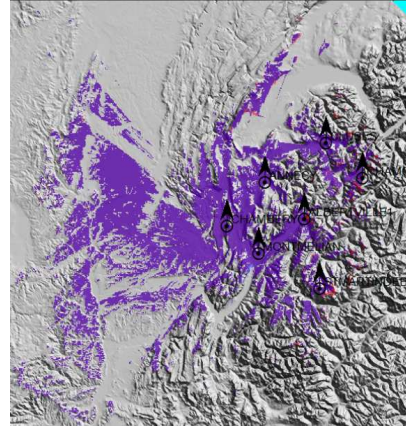
The densification of TV broadcasting networks and deployment of new radiocommunication services require an optimal use of the radio spectrum. The frequencies planning process determines a suitable frequency for each transmitter, so that no harmful interference arises in its coverage area. This process relies firstly on constraints modeling between transmitters, and secondly on the resolution of the corresponding Frequency Allocation Problems (FAP). In our constraints modeling, we consider binary interference constraints, i.e. involving only two transmitters. We consider also n-ary constraints due to SFN¹ networks. Satisfaction of these n-ary constraints is required to ensure the consistency of the frequency plan. Because of the strong links between graph coloring and frequency allocation with binary interference constraints, most methods found in the literature are based on graph coloring algorithms. The graph coloring problem is known to be NP-hard, thus, consequently the FAP problem. Therefore, exact methods are (in general) unable to solve large instances and heuristic approaches are necessary.

In this paper, we extend the single objective optimization version of the FAP problem to the multi-objective case (i.e. we associate an objective function to each network) and the task is now to find a set of various frequency assignments in a single run.

To solve this problem we have used an Ant Colony Optimization (ACO) algorithm [1] combined with Simulated Annealing algorithm [2] to improve the solution of each ant. Application to TDF's² real world benchmarks shows improvement of the quality of solutions as shown in next figures. Figure (a) corresponds to the best solution currently available and Figure (b) corresponds to the solution obtained by our approach. The comparison of these solutions shows that our algorithm obtains a better solution with less interference (red color).



(a)



(b)

References

- [1] É. D. Taillard, "FANT: Fast ant system", Technical report IDSIA-46-98, IDSIA, Lugano, 1998.
- [2] S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi, *Optimisation by simulated annealing*, Science, Vol. 220, pp. 671–680, 1983.

¹ SFN (Single Frequency Networks) is a set of synchronized transmitters, which broadcast the same bits of information using the **same** frequency. Due to its spectrum efficiency and spectrum reallocation to other services, SFN is widely used in Broadcasting networks

² TDF is a French company, which provides radio and television transmission services, services for telecom operators, and other multimedia services: digitization of content, encoding, storage, etc.

Hybrid PSO algorithm for the solving of the Optimal Reactive Power Problem

Zahir Sahli^a, Abdelghani Bekrar^b, Damien Trentesaux^b

^aDepartment of Electrotechnics, QUERE Laboratory
Ferhat Abbas Sétif 1 University, Sétif, Algeria
Sah_zah@yahoo.fr

^bUVHC, TEMPO lab, EA 4542, PSI Team, F-59313 Valenciennes, France

abdelghani.bekrar@univ-valenciennes.fr,
Damien.Trentesaux@univ-valenciennes.fr

Keywords: Mixed non-linear problem, Particle Swarm Optimization (PSO), hybrid method, Optimal Reactive Power Dispatch.

Abstract.

A power system is a complex network used for generating and transmitting electric power. It is expected to operate with consumption of minimal resources giving maximum security and reliability. The Optimal Power Flow (OPF) is an important problem to be solved to help the operator to achieve these goals (minimal resources consumption with maximum security and reliability) by providing the optimal settings of all controllable variables. Optimal Reactive Power Dispatch (ORPD) is a special case of OPF problem in which, control parameters are the variables which have a close relationship with the reactive power flow, such as generator bus voltages, output of static reactive power compensators, transformer tap-settings, shunt capacitors/reactors, etc[1]. Because of its significant influence on the secure and economic operation of power systems, ORPD has received an ever-increasing interest from electric utilities. The objective is to minimize the network real power loss and to improve the voltage profile, while satisfying a given set of operating and physical constraints. Because that outputs of shunt capacitors/reactors and tap-settings of transformers are discrete variables while other parameters in ORPD are continuous, the reactive power dispatch problem can be modeled as a mixed integer non-linear programming problem [2].

To solve the ORPD problem, meta-heuristics has been widely used [3]. Among these meta-heuristics, the Particle Swarm Optimization (PSO) is used in many papers, see for example [4,5,6]. The PSO was also hybridized with other method to avoid its premature convergence. However, a survey made on relevant published papers has

shown many lacks, especially regarding the implementation of the algorithm, the definition of the problem objectives and the considered constraints, the experimental design and the comparison of the proposed algorithms with other techniques.

In this paper, a critical overview of the papers using PSO to solve ORPD problem is first realized. A methodology to model this problem and to fix objectives and the different constraints is then defined and a new approach to solve the ORPD problem based on hybridizing the Particle Swarm Optimization and Tabu-search (PSO-TS) is proposed. This hybridization has never been used for the ORPD problem. We test and analyze different tuning of PSO-TS parameters. A comparative study is finally implemented on different IEEE network systems with other evolutionary algorithms to show the consistency and the performances of the proposed hybridization.

References

1. Alireza Abbasy Seyed Hamid Hosseini, "Ant Colony Optimization-Based Approach to Optimal Reactive Power Dispatch: A Comparison of Various Ant Systems", IEEE PES Power Africa 2007 - Conference and Exhibition Johannesburg, South Africa, 16-20 July 2007.
2. B. Zhao, C. X. Guo, and Y. J. Cao, "A Multiagent-Based Particle Swarm Optimization Approach for Optimal Reactive Power Dispatch", IEEE Transactions on Power Systems, Vol.20, NO.2, MAY 2005.
3. S. Frank, I. Steponavice, and S. Rebennack, « Optimal power flow: a bibliographic survey II », *Energy Syst*, vol. 3, n° 3, p. 259-289, sept. 2012.
4. Badar, A.Q.H., Umre, B.S., and Junghare, A.S. (2012). Reactive power control using dynamic Particle Swarm Optimization for real power loss minimization. *International Journal of Electrical Power & Energy Systems* 41, 133–136.
5. Hinojosa, V.H., and Araya, R. (2013). Modeling a mixed-integer-binary small-population evolutionary particle swarm algorithm for solving the optimal power flow problem in electric power systems. *Applied Soft Computing* 13, 3839–3852.
6. Mahadevan, K., and Kannan, P.S. (2010). Comprehensive learning particle swarm optimization for reactive power dispatch. *Applied Soft Computing* 10, 641–652.

An enhanced Particle Swarm Optimisation algorithm combined with Neural Networks to decrease computational time

Cédric Leboucher¹, Patrick Siarry², Stéphane Le Méné¹, Hyo-Sang Shin³,
Rachid Chelouah⁴, and Antonios Tsourdos³

¹ MBDA France

1, avenue Réaumur, 92350 Le Plessis Robinson, France

² Univ. de Paris-Est Créteil, Laboratoire Images, Signaux et Systèmes Intelligents,
LiSSI (E.A. 3956)

122 rue Paul Armangot, 94400 Vitry sur Seine, France

³ Cranfield University,

College Road, Cranfield, Bedfordshire, MK43 0AL, United Kingdom

⁴ EISTI

Avenue du Parc, 95000 Cergy, France

Abstract. This paper proposes to reduce the computational time of an algorithm based on the combination of the Evolutionary Game Theory (EGT) and the Particle Swarm Optimisation (PSO), named C-EGPSO, by using Neural Networks (NN) in order to lighten the computation of the identified heavy part of the C-EGPSO. This computationally burdensome task is the resolution of the EGT part that consists in solving iteratively a differential equation in order to optimally adapt the direction search and the size step of the PSO at each iteration. Therefore, it is proposed to use NN to learn the solution of this differential equation according to the initial conditions in order to gain a precious time.

Keywords: Particle Swarm Optimisation, Swarm Intelligence, Evolutionary Game Theory, Neural Networks

1 Introduction

Over the last few decades, numerous scientists have been inspired by the modelling of social interactions of animals to solve NP-hard optimisation problems. Although the communication among the different agents is limited to an exchange of basic information, it results in a very effective team work. Particle Swarm Optimisation (PSO) is one of most well-known and established approaches using this concept. The aim of the original PSO method proposed by Kennedy and Eberhart was to reproduce this social interaction among agents in order to solve non-linear continuous optimisation problems [1], [2]. PSO not only provides efficient and satisfactory solutions like other meta-heuristic methods [3], [4], [5], but also achieves more accurate results than traditional methods,

as Genetic Algorithms (GA) [6], [7], [8], for the problems involving unconstrained continuous functions and also more complex and highly constrained problems [9].

The PSO principle is based on sharing simple information such as current fitness, best obtained fitness and the best global fitness among neighbouring particles in order to determine moving rules of a swarm of candidate solutions, named particles. The movements of the particles are based on a random linear combination of their own current velocity, and the relative position vectors of their own best position and the best known position of the neighbouring particles with respect to their current position. Random choice of the three weighting parameters of the linear combination keeps diversification of the particles' search.

Based on the principle that these directions could be identified as strategies of exploration, the PSO was combined with Evolutionary Game Theory (EGT) in order to improve the convergence speed as well as the efficiency of the proposed method to solve complex optimisation problems. This method was named C-EGPSO and outperformed the Standard PSO 2011 [10] in terms of capacity to solve complex and various kinds of problems as well as the number of iterations required to converge to the optimal solution [11].

From this observation, it was devised to adapt the proposed C-EGPSO to real-time oriented applications. Since the EGT part consists in determining an Evolutionary Stable Strategy (ESS) iteratively by solving a system of Ordinary Differential Equations, the process was identified as a computationally heavy part of the method. Thus, this paper proposes the use of Neural Network to learn the output of the EGT process and predict the optimal strategy that optimises the exploration of the solution space.

The validation of the proposed approach will be done in two stages. First the validity of the NN will be verified by analysing the accuracy of the obtained results in the learning of the resolution of the Replicator equation, then the gain of computational time using NN will be quantified. Second, the proposed approach will be applied to a set of benchmarked continuous optimisation problems issued from [12] in order to confirm that the proposed algorithm obtains as good results as the original algorithm (C-EGPSO) in terms of precision of the final solution, as well as decreases the computation time for solving these problems. The second stage of the numerical validation will be done with a selection of benchmark functions issued from the CEC'2005 congress [12]. The benchmark definition is available in [13]. The efficiency of the proposed method was already tested in one of our previous studies ⁵.

The proposed paper will be organised as follows: first it will be provided to the reader the essential notions to understand the global approach, then the proposed method will be described, before giving details on the experimental procedure used in order to validate the approach. Finally, the obtained results will be presented and discussed. This paper will end with a conclusion and the perspectives of the presented approach.

⁵ *An Enhanced Particle Swarm Optimisation Method Integrated With Evolutionary Game Theory*. Submission in progress

References

1. J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, (Piscataway (USA)), pp. 1942 – 1948, 1995.
2. J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm," in *The 1997 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, (Orlando (USA)), pp. 4104–4108, October 1997.
3. W. Hu, J. Song, and W. Li, "A new PSO scheduling simulation algorithm based on an intelligent compensation particle position rounding off," in *ICNC '08 Proceedings of the 2008 Fourth International Conference on Natural Computation*, vol. 1, (Jinan (China)), pp. 145–149, October 2008.
4. U. Junker, "Air traffic flow management with heuristic repair," *The Knowledge Engineering Review*, vol. 0, pp. 1–24, 2004.
5. D. Shaa and L. H.H, "A multi-objective PSO for job-shop scheduling problems," *Expert Systems with Applications*, vol. 37, pp. 1065–1070, 2010.
6. M. Badamchizadeh and K. Madani, "Applying modified discrete particle swarm optimization algorithm and genetic algorithm for system identification," in *Computer and Automation Engineering (ICCAE), 2010*, vol. 5, (Singapore (Republic of Singapore)), pp. 354–358, February 2010.
7. R. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Proceedings of the 7th International Conference on Evolutionary Programming*, (San Diego (USA)), pp. 611–616, March 1998.
8. C. Liaoa, C. Tsengb, and P. Luarnb, "A discrete version of particle swarm optimization for flowshop scheduling problems," *Computers & Operations Research*, vol. 34, pp. 3099–3111, 2007.
9. R. Hassan, B. Cohanin, O. de Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," tech. rep., Massachusetts Institute of Technology, Cambridge, MA, 02139, 2004.
10. M. Clerc, "Standard particle swarm optimisation," tech. rep., <http://clerc.maurice.free.fr/>, 2012.
11. C. Leboucher, H.-S. Shin, P. Siarry, R. Chelouah, S. L. Menec, and A. Tsourdos, *A Two-Step Optimisation Method for Dynamic Weapon Target Assignment Problem, Recent Advances on Meta-Heuristics and Their Application to Real Scenarios*. InTech, 2013.
12. *Congress on Evolutionary Computation*, (Edinburgh (UK)), September 2005.
13. 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," tech. rep., Technical Report, Nanyang Technological University, Singapore and KanGAL Report Number 2005005 (Kanpur Genetic Algorithms Laboratory, IIT Kanpur), 2005.

An hybrid PSO applied to Flexible Manufacturing Systems

Laurent Deroussi

LIMOS UMR-CNRS 6158, Avenue des Landais, 63173 Aubière Cedex, France
laurent.deroussi@moniut.univ-bpclermont.fr

Abstract. In this paper, we study the hybridization of particle swarm optimization (PSO) with stochastic local search applied to a difficult scheduling problem, recently appeared in the literature: the flexible job shop problem with transport. The objective is to determine experimentally the best balance between the exploration ability of PSO and the exploitation ability of the local search.

Keywords: Particle Swarm Optimization, hybrid metaheuristics, flexible manufacturing systems, flexible job shop problem, automated guided vehicles.

Introduction

In this paper, we study the hybridization of particle swarm optimization (PSO) with stochastic local search applied to a difficult scheduling problem, recently appeared in the literature: the flexible job shop problem with transport. The objective is to determine experimentally the best balance between the exploration ability of PSO and the exploitation ability of the local search.

Description of the hybrid PSO

General framework

Particle Swarm Optimization (PSO) is a nature inspired metaheuristic developed by (Eberhart & Kennedy, 1995) for continuous optimization problems. However, many variants have been proposed for discrete optimization problems. The framework of PSO can be described as follows:

- At iteration k , each particle i has a position $x_{i,k}$ that represents a solution of the optimization problem. The objective function gives the cost of this solution and defines the fitness of the particle.
- Each particle i remembers the best solution $pbest_i$ it has encountered in its past (local memory)

- Each particle knows the best solution $gbest_i$ encountered either by the whole swarm, or by a subset of it (global memory).
- Each particle flies in the solution space with a given velocity $v_{i,k}$.

Starting from a swarm of random particles (position and velocity), each of them will move in the solution space using the two following equations:

$$v_{i,k+1} = \omega \times v_{i,k} + c_1 \phi_1 (pbest_i - x_{i,k}) + c_2 \phi_2 (gbest_i - x_{i,k}) \quad (1)$$

$$x_{i,k+1} = x_{i,k} + v_{i,k+1} \quad (2)$$

The hybrid PSO

As many other population metaheuristics, PSO is a powerful metaheuristic in terms of exploration. The idea is to combine PSO with a local search procedure which is powerful in terms of exploitation. The pseudo-code of the hybrid PSO is given in algorithm 1:

```

Algorithm 1: pseudo-code of hybrid PSO
Generate the initial swarm (position  $x_{i,0}$  and velocity  $v_{i,0}$ )
Initialize the local and global best positions  $pbest_i$  and  $gbest_i$ 
For each generation k
  For each particle i
    Compute the velocity  $v_{i,k}$  using equation (1)
    Update the position  $x_{i,k}$  using equation (2)
    Improve  $x_{i,k}$  with local search
    If necessary, update  $pbest_i$ 
  End for i
  If necessary, update  $gbest_i$ 
End for k

```

The objective of this hybrid metaheuristic is to combine their complementary strengths and weaknesses in order to improve its efficiency.

The stochastic local search

We propose to embed into PSO a stochastic variant of the local search. The general framework of local search is to improve a solution x by searching into a subset of solution denoted $N(x)$ and called the neighborhood of x . The pseudo-code is given in algorithm 2.

```

Algorithm 2: pseudo-code of stochastic local search
Generate an initial solution  $x$ 
Initialize Max_failure

```

```

nb_failure:=0
While nb_failure<Max_failure do
  Generate at random  $y \in N(x)$ 
  If  $c(y) < c(x)$  then
     $x := y$ 
    nb_failure:=0
  else
    nb_failure:=nb_failure+1
  end if
end while
return  $x$  the pseudo local minimum

```

A high value for the parameter *Max_failure* indicates that a lot of tries without improvement are allowed before the local search stops. This means that if a better solution y exists in the neighborhood of x , the probability p to find it increases with *Max_failure*. More precisely, the number of tries n required for finding y with probability p can be estimated by the following formula:

$$n \approx -\ln(1-p) \times \text{Card}(N) \quad (3)$$

In practice, it is often recommended to fix $\text{Max_failure} = 0.7 \times \text{Card}(N)$. This value corresponds to a probability $p \approx 0.5$.

Our objective is to use the parameter *Max_failure* in order to increase or decrease the exploitation impact of the local search into the hybrid PSO.

The flexible job shop problem with transport

The problem under consideration consists in the simultaneous solution of the scheduling of AGVs and the scheduling of machines in an FMS environment. The FMS is considered to be of the flexible job shop type. Each job consists of a set of ordered operations. Each operation needs to be processed during a given amount of time on a given type of machines. A type of machine characterizes a subset of the machines (identical or not). Each machine can handle at most one operation at a time. Each job has to be transported from an origin machine to a destination machine by an AGV. The calculation of the makespan requires synchronization between the production tools and the material handling system.

This problem is appeared recently in the scheduling literature. It is the combination of two problems: the job-shop with transport (introduced by Ulusoy & Bilge, 1995) and the flexible job-shop problem (introduced by Brandimarte, 1993).

References

Bilge, U, & Ulusoy, G. (1995). A time window approach to simultaneous scheduling of machines and material handling system in an FMS. *Operations Research*, 43, 1058-1070.

Brandimarte, P., (1993). Routing and Scheduling in a Flexible Job Shop by Tabu Search. *Annals of Operations Research*, 41, 157–183.

Eberhart, RC., & Kennedy J. (1995). A new optimizer using particle swarm theory. Paper presented in the sixth international symposium on micro machine and human science, IEEE service center. Piscataway, NJ, Nagoya, Japan.

A study of task division of ants for online tuning of metaheuristic parameters

Nicolas Monmarché, Sébastien Aupetit, Pierre Gaucher, Mohamed Slimane

University François Rabelais, Tours, Laboratoire d'Informatique,
64 Av. Jean Portalis, 37200 Tours, France

Abstract. This paper presents a study of how a task division model in ants can be used to provide a self adaptation mechanism in a global optimization algorithm.

It is fascinating to observe that ants, particularly in species having large population, can display specialized behaviors. One fascinating point is that the more they are specialized in a reduced set of tasks (nest maintenance, food gathering, brood care...) the more they spend time to rest. But, when the need in their field is increasing they are very quickly responding to the demand and they are efficient.

The main point of the presented work is to study how the task division mechanism, i.e. the ability of ants to adapt their work force to the needs of the colony, can be employed as a model of self adaptation of the work of ants that are performing an optimization task.

In a previous study [5], we have already focused our attention to this task division model, but for a completely different application field: ants were used to self adapt to a human player's skills in a small game designed for a wide audience (i.e. also for disable persons).

The global optimization algorithm we consider here is also an ant algorithm based on real ants of *Pachycondyla apicalis* species. This algorithm, namely the API algorithm [4], is chosen because of its simplicity to be understood quickly but not because of its performances since several more recent algorithms perform better. The API algorithm can be considered as simple because the ants behaviors on which it relies on are also simple: the *Pachycondyla apicalis* ants do not use any pheromones to share information between them and the individual ant's strategy is simple. In previous research works, we have studied how API algorithm can compete with other population based algorithms, particularly on learning statistical models [3, 1, 2] but in the scope of this paper, we are focusing on standard test functions defined on continuous domains.

References

1. Sébastien Aupetit, Nicolas Monmarché, and Mohamed Slimane. The use of hidden markov models for image recognition: learning with artificial ants, genetic algorithms and particle swarm optimization. In Patrick Siarry, editor, *Optimization in Signal and Image Processing*. ISTE - Wiley, June 2009.

2. Sébastien Aupetit, Nicolas Monmarché, and Mohamed Slimane. Training of hidden markov models using api ant algorithm. In Nicolas Monmarché, Frédéric Guinand, and Patrick Siarry, editors, *Artificial Ants: from collective intelligence to real life optimization and beyond*, chapter 12. ISTE - Wiley, 2010.
3. Sébastien Aupetit, Nicolas Monmarché, Mohamed Slimane, and Pierre Liardet. An Exponential Representation in the API Algorithm for Hidden Markov Models Training. In E. Talbi et al., editor, *Artificial Evolution (EA '2005)*, volume 3871 of *Lecture Notes in Computer Science*, pages 61–72, Lille, France, 2006. Springer-Verlag Berlin Heidelberg. (revised selected papers).
4. Nicolas Monmarché, Gilles Venturini, and Mohamed Slimane. On how *Pachycondyla apicalis* ants suggest a new search algorithm. *Future Generation Computer Systems*, 16(8):937–946, 2000.
5. Alexis Sepchat, Romain Clair, Nicolas Monmarché, and Mohamed Slimane. Task division in ants for better game engines: a contribution to game accessibility for impaired players. In G. Rudolph, Th. Jansen, S.M. Lucas, C. Poloni, and N. Beume, editors, *Parallel Problem Solving from Nature - PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 961–970. Springer Berlin / Heidelberg, Dortmund, Germany, September 13-17 2008.

Using bio-inspired algorithm to compensate web page color contrast for dichromat users

A. Mereuta, S. Aupetit, N. Monmarché, and M. Slimane

Université François Rabelais Tours, Laboratoire Informatique (EA6300), France
64, avenue Jean Portalis, 37200 Tours, France
{alina.mereuta,aupetit,monmarche,slimane}@univ-tours.fr

Abstract. With this paper we are focusing on improving web accessibility, more precisely on compensating the contrast loss for textual web content for dichromat users. A study over the entire sRGB color space showed that the loss experienced by a dichromat user may be significant. With the current approach, we assess the interest of using API for our problem. Several tests for different parameters settings were performed on both real and synthetic data in order to assess the algorithm efficiency.

Keywords: dichromacy, assistive technologies, API, accessibility, recoloring, optimization, swarm intelligence

1 Introduction

Accessibility concern is to ensure the same level of availability of resources for both standard and disabled people. Web accessibility, in particular, focuses on ensuring unlimited web access for users with disabilities. Towards improving web accessibility, several steps were made by developing standards and policies. Guidelines that may serve as references for developing accessible web sites may be considered Web Content Accessibility Guidelines (WCAG) 1.0¹ and 2.0, proposed by Web Accessibility Initiative (WAI), a World Web Consortium (W3C) working group. Web site compliance with such policies may increase the degree of accessibility of a web site. However, very few web designers check for accessibility issues their web sites. This results in web content that can be difficult to access by users having some form of disability. To address this, a series of transformation tools that attempt to correct these shortcomings were proposed [9, 4, 3].

A form of deficiency is Dichromacy. It is a color deficiency which makes difficult for a person to discriminate between certain colors. This is due to the lack of one of the three types of cone cells in eye's retina which make possible the perception of colors. Corresponding to the range of misperceived shades of colors, two main categories of dichromacy exist: (1) red-green deficiencies (protanope (missing L cones) and deuteranope (lack of M cone cells)) and (2) blue-yellow deficiency (tritanope (S cones are missing)). To address this issue

¹ <http://www.w3.org/TR/WCAG10/>

a series of algorithms and tools were proposed. To simulate color deficiencies several algorithms [1, 2, 6, 11] and tools ² were developed. Many attempts to diminish the shortcomings induced by this type of deficiency exist [12, 5, 10].

In the following, we are focusing on compensating color contrast for web textual information for dichromats users.

2 The problem

WCAG 1.0 structures accessibility standards into four main groups: understandable, perceivable, robust and operable. For all of them it introduces a three steps way to measure web page accessibility from Level A to AAA. Level A corresponds to the minimum of accessibility required for a web site. One requirement concerning the representation of textual information on the page and more precisely , concerning the color contrast, (Guideline 2) states the following “Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen”.

WCAG provides ways to measure the contrast ratio for textual information on the web. Let $a, b \in [0, 255]^3$ be two colors represented in the sRGB color space (known as the Internet standard color space). The luminance of $x \in [0, 255]^3$ is defined as:

$$L(x) = 0.2126 * h(x^r) + 0.7152 * h(x^g) + 0.0722 * h(x^b) \quad (1)$$

with

$$h(z) = \begin{cases} \frac{z/255}{12.92} & \text{if } z/255 \leq 0.03928 \\ \left(\frac{z/255 + 0.055}{1.055} \right)^{2.4} & \text{otherwise} \end{cases} . \quad (2)$$

The contrast between a and b is given by:

$$\Gamma_{a,b} = \frac{\max(L(a), L(b)) + 0.05}{\min(L(a), L(b)) + 0.05} \in [1 : 21] . \quad (3)$$

We denote by $D(a) \in [0 : 255]^3$ the function used to obtain the corresponding color as perceived by a dichromat user and by $\Gamma_{a,b}^D$ the simulated contrast for the colors a and b . The simulation algorithm used is the one proposed by [6].

A small study on contrast ratio revealed that the contrast loss experienced by a dichromat user may be significant. At its worst, it can reach a decrease of around 3.8 for protanope or deuteranope and 3.7 for tritanope, while the minimum level specified by the standards (WCAG 1.0) is 4.5:1.

In this work, we attempt to compensate the textual contrast loss that might be experience by a dichromat user.

Let \mathcal{C} be the set of colors that may be found on an arbitrary web page and $\mathcal{E} \subset \mathcal{C} \times \mathcal{C}$ the set of entities characterized by foreground and background color.

² <http://www.vischeck.com/>

Let be $\Delta_i = |a_i - a_i^F|_{CIELab}$, $\forall a_i \in \mathcal{C}$ the Euclidean distance between the original and the transformed color

The contrast compensation problem can be modeled as a mono-objective function given by:

$$F(a_1^F, a_2^F, \dots, a_N^F) = (1-\alpha) \sum_{(a_i, a_j) \in \mathcal{E}} \frac{1}{2} \left[\max(\Gamma_{i,j}^I - \Gamma_{i,j}^{F,D}, 0) \right]^2 + \alpha \sum_{c_i \in \mathcal{C}} \frac{1}{2} \Delta_i^2 \quad (4)$$

that aims:

1. to compensate the contrast loss by minimizing $\max(\Gamma_{i,j}^I - \Gamma_{i,j}^{F,D}, 0)$, $\forall (i, j) \in \mathcal{E}$
2. to reduce the change in the final transformed colors by minimizing (Δ_i)

We also use a constant α to weight between the amount of compensation needed and the change in colors. The compensation contrast problem may be reduced at minimizing (4).

In previous work, we have considered a mass-spring approach and the CMA-ES algorithm [7] to solve the problem. The results on both real and artificial data were encouraging. In the following we are investigating the interest of using the API algorithm for solving our problem.

3 The proposed approach

For the experiments, we are using the proxy part of the SWAP(Smart Web Accessibility Platform)³. The platform has as main goal to improve web accessibility for disable users. In this work, we are only using the proxy part of SWAP which allows on-the-fly transformation of web pages when the user access Internet. SWAP handles colors extraction from the CSS of the page and the inclusion of the changes in the web page set to the client browser.

The goal with this paper is to assess the efficiency of the API algorithm for our problem. The API algorithm [8] is based on the foraging strategy of the *Pachycondyla apicalis* ants. It allows to minimize a function f in a search space S . In the following we recall its main principles.

Let be $\{1, 2, \dots, n\}$ an ant colony and $\{s_i\}_{1,n}$ a set of hunting sites, associated to the ant colony (each ant has an hunting site). The algorithm behaviour is based on two operators:

1. \mathcal{O}_i is used to initialize the colony nest (\mathcal{N}) (in our approach it will be given by the set of initial colors)
2. $\mathcal{O}_e(x, \mathcal{A})$ is used to generate a solution in the neighborhood of x . The neighborhood size depends on amplitude, \mathcal{A} .

Two sets of amplitudes are needed: one for the hunting sites ($\{\mathcal{A}_i^s\}$) and one that gives the extend to which a hunting site will be explored ($\{\mathcal{A}_i^l\}$). In the following

³ <http://projectsforge.org/projects/swap>

we consider them to be given by: $\mathcal{A}_i^s = 1 + (0.01 - 1)^{\frac{i-n}{1-n}}$ and $\mathcal{A}_i^l = \mathcal{A}_i^s/10$ as defined initially by N. Monmarche. To control the number of failures for an hunting site a counter and a maximum limit are defined.

The algorithm behaves as following: a hunting site is assigned to each ant from the colony if they don't have one using $\mathcal{O}_e(\mathcal{N}, \mathcal{A}_i^s)$, if they have one a search for a solution is done with $(\mathcal{O}_e(s_i, \mathcal{A}_i^l))$. If the solution found is better than the current solution represented by the hunting site, the hunting site is moved to the new best solution and the counter of failures is brought to zero. If the explored solutions are worst than the current one (the hunting site) the failure counter is increased. If the maximum failure counter is attained for a hunting site, the ant abandons it. Regularly the nest is moved in the best solution ever found and all the hunting sites are abandoned.

In the following experiments were conducted in order to evaluate the interest of using it for solving our problem, on both real and synthetic data.

4 Results and discussion

A set of experiments were conducted in order to assess the behaviour of the API algorithm on the given problem (color optimization for dichromat users). Tests were performed on two types and data: real and synthetic. Real data was obtained by from CSS analysis on over 170 pages using SWAP. Concerning the synthetic data, the colors were generated according to the given confusion range used in [2] (the pairs of colors that fall in that specific range are perceived similar by a dichromat user, so the contrast ratio perceived will be much smaller than the one perceived by a standard user). 170 synthetic data files were computed

For API, we have considered many parameters settings (PS), varying the maximal limit for the number of failures for a given hunting site $\in \{5, 10, 15\}$, the number of explored solutions by iteration $\in \{5, 10, 15\}$ and the maximal limit for the number of failure before changing the hunting site $\in \{10, 20, 30\}$. The same settings were maintained for both real and synthetic data.

For each web page, 10000 evaluation were performed using the API algorithm, for a number of 50 tries, for all three types of parameter settings. We denote by $\bar{F}_{p,s}(t)$ the average fitness value over the number of tries at time t , for the page p corresponding to the parameter setting s . In order to assess the parameter settings behaviour we have normalized $\bar{F}_{p,s}(t) \forall t$ as follow:

$$g_{p,s}(t) = \frac{1}{|\mathbb{P}|} \sum_{p \in \mathbb{P}} \frac{F_{p,s}(t) - m_p}{M_p - m_p} . \quad (5)$$

where g is the normalized average fitness values for the page p and parameter setting s , m_p and M_p represent the minimum, respectively the maximum value for $\bar{F}_{p,s}$ for the entire set of parameter settings considered for the page p . \mathbb{P} represents the dataset considered, in our case, is either *RD* (real data) or *SD* (synthetic data).

Experiments showed that the algorithm behaviour does vary with the change in the chosen parameters. The performance is better for the *PS1*(5, 5, 10) compared to the others on real data. The last two perform similary on real data,

slightly better for *PS2*. On synthetic data the behaviour is very similar for all types of PS and all types of deficiencies.

The first parameter setting, *PS1*(5, 5, 10) gives the best results for both real and synthetic data as depict by Figure 1 and 2. This may be due to the fact that by limiting the number of failures at hunting site level and at nest level it has the oportunity to explore a larger area of the search space and that allows to find rapidly, better solutions.

It worth meantioning, that API behaviour depends highly on the dataset. It can produce less than satisfactory results also, dependind on the data as seen in Figure 3.

We can notice that API performs better on synthetic data than on real data.

We are interested also to determine if the algorithm can be used for on the fly page recoloring. This may be translated by the possibility of obtaining a good improvement over a small number of evalutation and that the time to performe that number of evaluations is nor significant.

Table 1 shows the normalized average performance for 500, 1000, 1500 and 2000 evaluations.

As we can notice in Table 1 more than half of the total improvement can be achieved for less than 500 evaluations for real data and almost all the improvement for synthetic data.

The time to recolor a page varies with: (1) DOM retrival, (2) CSS analysis , (3) color optimization and sending the modified page with the new colors to the browser).

5 Conclusion

With this paper, we have assessed the interest of using API algorithm for the problem of contrast compensation for dichromat user. We have obtained good results on both real and synthetic dataset for all three types of dichromacy considered. Several tests still need to be performed in order to determine if the algorithm is more suitable than our previous approaches to be used for on-the-fly recoloring of web pages.

References

1. Brettel, H., Vienot, F., Mollon, J.: Computerized simulation of color appearance or dichromats. *Journal of Optical Society of America* 14(10), 2647–2655 (1997)
2. Brettel, H., Vienot, F., Mollon, J.: Digital video colourmaps for checking the legibility of displays by dichromats. *Color Research and Application* 24(4), 243–251 (1999)
3. D. Lunn, S.H., Bechhofer, S.: Combining SADie and AxsJAX to Improve the Accessibility of Web Content. In: *Proceedings of W4A2009 Communication*. pp. 75–78. April Madrid, Spain, Madrid (20-21 April 2009)
4. Gupta, S., Kaiser, G.: Extracting Content from Accessible Web Pages. In: *Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A’05)*. pp. 26–30. New York, USA, 2005 (2005)

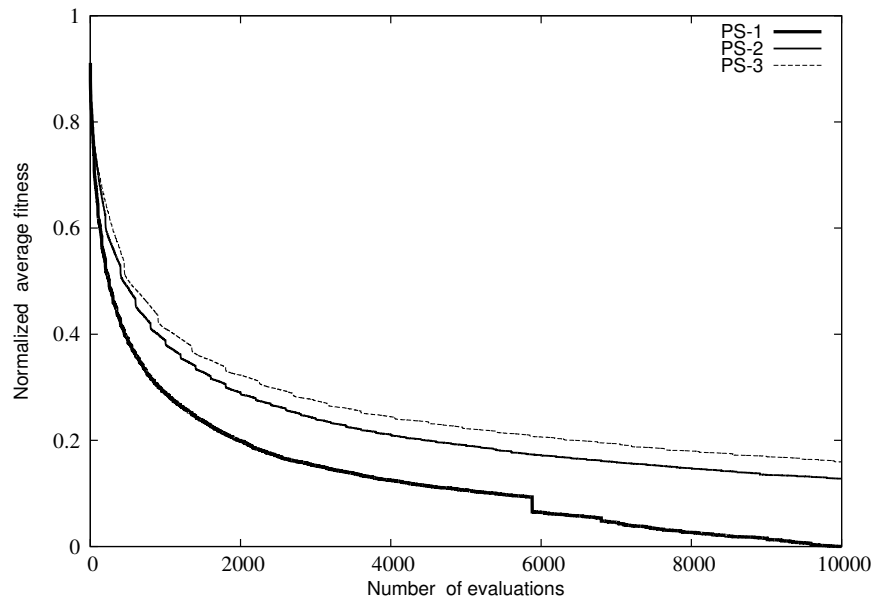


Fig. 1: Normalized average fitness for Deuteranope on real data for PS1, PS2 and PS3

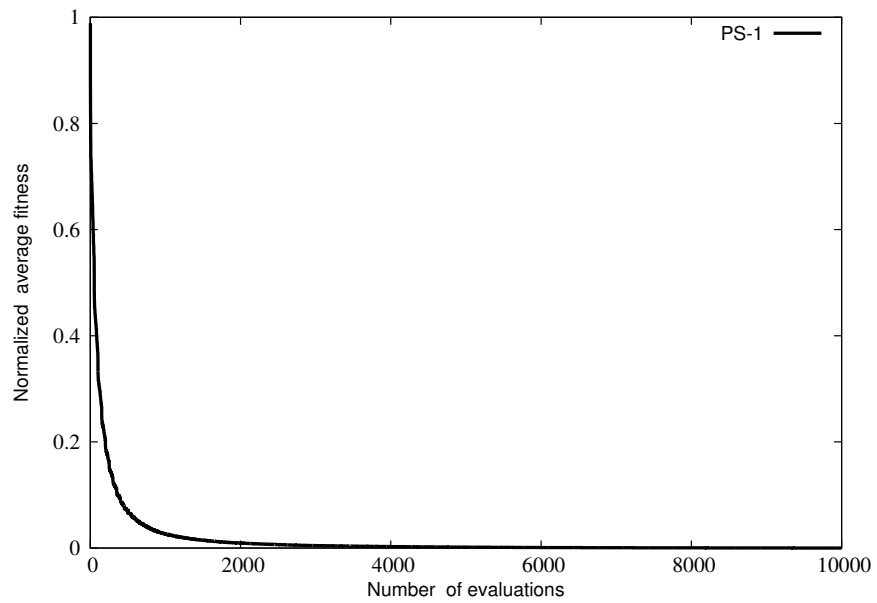


Fig. 2: Normalized average fitness for Tritanope on synthetic data for PS1

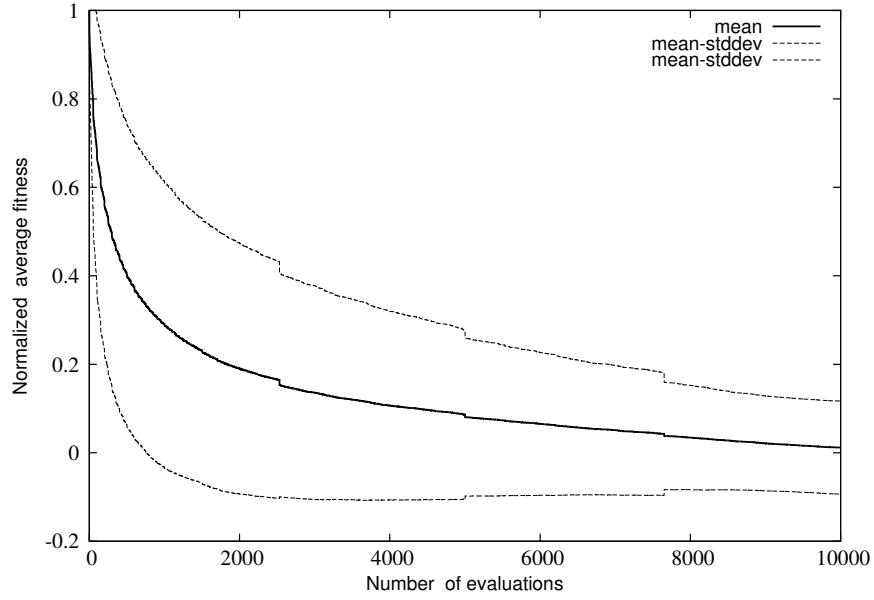


Fig. 3: Normalized average performance for Protanope on real data for PS1

Table 1: Normalized average fitness at 500, 1000, 1500 and 2000 evaluations for PS1 for Protanope, Deuteranope and Tritanope on real(RD) and synthetic(SD) data.

<i>Type of deficiency</i>	<i>RD</i>					<i>SD</i>				
	500	1000	1500	2000	2500	500	1000	1500	2000	2500
Deuteranope	0.39	0.29	0.24	0.20	0.17	0.08	0.03	0.02	0.01	0.01
Protanope	0.41	0.29	0.23	0.19	0.17	0.08	0.03	0.02	0.01	0.01
Tritanope	0.42	0.32	0.26	0.22	0.20	0.07	0.03	0.01	0.01	0.01

5. Iaccarino, G., Malandrino, D., et al.: Efficient Edge-Services for Colorblind Users. In: WWW '06 The 15th international conference on World Wide Web. pp. 919–920. Edinburgh, Scotland UK (May 22 - 26 2006)
6. Kuhn, G.R., Oliveira, M.M., Fernandes, L.A.F.: Efficient Naturalness-Preserving Image-Recoloring Method for Dichromats. *Visualization and Computer Graphics*, IEEE 6(14), 1747–1754 (2008)
7. Mereuta, A., Aupetit, S., Monmarch, N., Slimane, M.: Web page textual color contrast compensation for CVD users using optimization methods. *Journal of Mathematical Modelling and Algorithms in Operations Research* (Oct 2013)
8. Monmarché, N., Venturini, G., Slimane, M.: On how it *Pachycondyla apicalis* ants suggest a new search algorithm. *Future Generation Computer Systems* 16(8), 937–946 (2000)
9. Parmanto, B., Ferrydiansyah, R., Zeng, X., Saptono, A., Sugiantara, I.W.: Accessibility Transformation Gateway. In: HICSS: Proceedings of the 38th Annual Hawaii International Conference on System Sciences. pp. 3–6. Island of Hawaii, Hilton Waikoloa Village (January 2005)
10. Rodriguez-Pardo, C.E., Sharma, G.: Adaptive Color Visualization for Dichromats using a Customized Hierarchical Palette. *Color Imaging XVI: Displaying, Processing, Hardcopy, and Applications*, Volume 7866 (2011)
11. Ruminski, J., Wtorek, J., Ruminska, J., Kaczmarek, M., Bujnowski, A., Kocejko, T., Polinski, A.: Color transformation methods for dichromats. pp. 634–641. IEEE, <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5514503>
12. Wakita, K., Shimamura, K.: SmartColor: Disambiguation Framework for the Colorblind. In: ASSETS'05, The sixth International ACM Access Conference on Assistive Technologies. pp. 158–165. Baltimore, USA (October 9-12 2005)

Comparison of two swarm intelligence optimization algorithms on the textual color problem for web accessibility

Sébastien Aupetit, Nicolas Monmarché, and Mohamed Slimane

Université François Rabelais Tours, Laboratoire Informatique (EA6300)
64, avenue Jean Portalis, 37200 Tours, France
{aupetit,monmarche,slimane}@univ-tours.fr

Abstract. Currently, web accessibility is not a major concern of webmasters while creating web sites. For disabled people, it rapidly becomes an obstacle to inclusion in the society. Identifying and circumventing existing barriers constitute an important research topic. In this work, we are concerned with the problem of color accessibility of textual contents in web pages. In many cases, the textual colors of a web page do not respect the minimum constraints defined by recommendations like WCAG 2.0. For example, WCAG 2.0 requires that a minimum difference of brightness, tonality and contrast is ensured. Using the Smart Web Accessibility Platform, we try to transform the colors using a client-side HTTP proxy the best possible while retaining a reasonable access time for the web content. To solve the textual color problem for accessibility, we adapt two swarm intelligence based optimization methods (ABC and API) and we hybridize them with a line search.

Keywords: accessibility, assistive technology, recoloring, web, swarm intelligence based optimization, ABC, API

1 Introduction

Web accessibility is a big concern for disabled people when accessing Internet. While Internet contributes to the insertion in the society, it can leads to exclusion when it is not accessible. To reduce exclusion, active or passive accessibility can be used. Active accessibility consists in a pro-active approach relying on norms, recommendations¹, laws [1], tools² and methodologies [2] to enforce a proper structuring and tagging of documents during their creation by webmasters. While it is the ideal way for achieving accessibility, in practice, a fully accessible web is not possible mainly due to limited implication of webmasters or external constraints (money, time. . .). Moreover, many web sites are unmaintained or ageing. Passive accessibility consists in using assistive technologies and transformation tools to allow a better access to web contents. This form

¹ <http://www.w3.org/TR/WCAG10/>, <http://www.w3.org/TR/WCAG20/>

² <http://achecker.ca/checker>, http://www.binaryblue.com.au/access_wizard...

of accessibility is our concern in the project Smart Web Accessibility Platform (SWAP)³ that we develop. It is a set of open source modular components and tools designed to facilitate content transformation (the proxy tool), to store and to share global knowledge and to create metadata (annotations...) on web pages. A more detailed description of the project can be found in [3]. In this work, we are concerned only by the proxy tool. It is located on the user computer and is used by the browser to access Internet. On request of the browser, the proxy fetches the content on Internet, applies transformations on the page and sends the modified page to the browser. It allows to transform any content passing by the proxy (even secured ones) and to preserve existing user assistive technologies (speech synthesis, zoom...) so increasing the adoption of our tool. The transformations do not aim to make the content accessible for the average disabled user but for specific user needs: we transform only what the user wants and needs. The transformations are done on the fly while the page passes by the proxy. For ergonomic reasons, the user can not wait too long. Moreover, a perfect improvement of the content is not possible due to the lack of information on the contents. Consequently, the transformation of the content using the proxy tool is a time limited and imperfect process. However, a partial improvement of the content is nevertheless a big improvement for the user.

In this work, we are focused on textual color improvement which can be formalized as an optimization problem. In previous works [4–7], we considered simple heuristics and metaheuristics to solve the problem and a fitness function prioritizing four measures and using an integral and a fractional parts formulation. This fitness function while allowing a huge improvement of the colors suffers a big practical defect. It introduces discontinuities leading to color schemes which are equally ranked but which can be very different. From a user perspective, the important changes on the color schemes, each time the user access the same web page, is an issue. From now, the fitness function is modified into a weighted function of the four measures on the color schemes. It produces more stable color schemes over time at the price of eventually reducing constraints satisfaction. In the following, we define the textual color problem, its specificities and how it can be solved with three swarm intelligence based optimization methods.

2 The textual color problem for accessibility

2.1 Textual colors accessibility

The accessibility definitions considered are defined by WCAG 1.0 and 2.0 (Web Content Accessibility Guidelines)¹. In the following, colors are considered in the sRGB space. To be accessible, the foreground (f) and background (b) colors of any text must satisfy three constraints: a minimal brightness ($\Delta B(f, b) \geq \eta_B$), a minimal tonality difference ($\Delta T(f, b) \geq \eta_T$) and a minimal contrast ($\Delta C(f, b) \geq \eta_C$). η_B , η_T and η_C are the accessibility thresholds fixed at 125, 500 and 7 in

³ <https://projectsforge.org/projects/swap>

the following. For any colors $x = (x_1, x_2, x_3)$ and $y = (y_1, y_2, y_3)$ defined on $\llbracket 0 : 255 \rrbracket^3$, we have:

$$\Delta B(x, y) = |0.299(x_1 - y_1) + 0.587(x_2 - y_2) + 0.114(x_3 - y_3)| \quad (1)$$

$$\Delta T(x, y) = |x_1 - y_1| + |x_2 - y_2| + |x_3 - y_3| \quad (2)$$

$$\Delta C(x, y) = \frac{\max(L(x), L(y)) + 0.05}{\min(L(x), L(y)) + 0.05} \quad (3)$$

$$L(x) = 0.2126h(x_1) + 0.7152h(x_2) + 0.0722h(x_3) \quad (4)$$

$$h(v) = \begin{cases} \frac{v/255}{12.92} & \text{if } v/255 \leq 0.03928 \\ \left(\frac{v/255 + 0.055}{1.055} \right)^{2.4} & \text{otherwise} \end{cases} \quad (5)$$

2.2 Distance between colors

We define $\Delta E(x, y)$ the classical perceptual distance in CIE L*a*b* divided by $100\sqrt{3}$. This distance is an euclidean distance in the CIE L*a*b* color space which measures the difference between two colors like what a human perceives it. To compute $\Delta E(x, y)$, the colors in the sRGB space must be transformed into colors in the CIE L*a*b* color space. For concision, details are not given here.

2.3 Objective function

Colors of textual contents are extracted from web pages parsing CSS (Cascading Style Sheets) and HTML contents. Identical colors in foreground are merged and the same is done for the background colors. We do it in order to preserve the coloring intention of the webmaster (visual identification...). Let $\mathcal{C} = \{c_1, \dots, c_{|\mathcal{C}|}\}$ be the set of colors used to represent the textual information of a web page. Let \mathcal{E} be the set of couples (foreground, background) presented in our page and $w_{x,y} \in \mathbb{R}^+$ the associated weights. In our modeling, these weights are the number of characters in the web page that use the couple of colors (x, y) . Let $c_x^I \in \llbracket 0 : 255 \rrbracket^3$ be the initial color coordinates for $x \in \mathcal{C}$ and $c_x^F \in \llbracket 0 : 255 \rrbracket^3$ the coordinates of the same color after the transformation. We define $S_w = \sum_{(x,y) \in \mathcal{E}} w_{x,y}$ the sum of the weights of the all couple of colors.

Interdependencies between colors can lead to unsatisfiable problems. To handle this issue, we relax the constraints using the Φ function as an evaluation of the constraint violation such that, for all $v \in [0 : M]$: $\Phi(v, T) = \max(0, (T - v)/T)$. If $v \geq T$ then $\Phi(v, T) = 0$. Otherwise, $\Phi(v, T)$ increases linearly until 1 when v decreases to 0 (the worst violation). We define :

$$S_B(c^F) = S_w^{-1} \sum_{(x,y) \in \mathcal{E}} w_{x,y} \Phi(\Delta B(c_x^F, c_y^F), \eta_B) \quad (6)$$

$$S_T(c^F) = S_w^{-1} \sum_{(x,y) \in \mathcal{E}} w_{x,y} \Phi(\Delta T(c_x^F, c_y^F), \eta_T) \quad (7)$$

$$S_C(c^F) = S_w^{-1} \sum_{(x,y) \in \mathcal{E}} w_{x,y} \Phi(\Delta C(c_x^F, c_y^F) - 1, \eta_C - 1) \quad (8)$$

$$S_E(c^F) = S_w^{-1} \sum_{(x,y) \in \mathcal{E}} w_{x,y} (\Delta E(c_x^I, c_x^F) + \Delta E(c_y^I, c_y^F)) / 2 \quad (9)$$

S_B , S_T and S_C measure the constraints violation while S_E measures the color difference perceived by an human between the initial colors and the new colors. Let ϵ_B , ϵ_T and ϵ_C be three weights defined by the user to weight the constraints according to his needs. In ours experiments, we considered $\epsilon_B = \epsilon_T = \epsilon_C = 1$. Two color schemes can satisfy the constraints to the same level (for example, $S_B(c^F) = S_T(c^F) = S_C(c^F) = 0$). To differentiate the schemes, we use S_E to prefer the scheme which changes the less the perceived colors (for example, we prefer changing pink into red instead of green). Let $\alpha \in]0 : 1]$ be an user definable weight to balance between the satisfaction of the constraints and the minimization of the colors change. In experiments, we considered $\alpha = 0.8$. The objective function is defined by:

$$F(c^F) = \alpha * \frac{\epsilon_B S_B(c^F) + \epsilon_T S_T(c^F) + \epsilon_C S_C(c^F)}{\epsilon_B + \epsilon_T + \epsilon_C} + (1 - \alpha) * \Delta E(c^F) \quad (10)$$

The textual color problem for web accessibility consists in minimizing $F(c^F)$. The computation must be accomplished on-the-fly while the content passes by the proxy. The available computation time depends on the user computer speed, on the running processes and on the other transformations applied to the content. We do not know *a priori* how much computation time is available so we consider that the minimization of F may be interrupted at any moment. The best found solution is used to recolor the page. Even if the recoloration is not optimal, it is nevertheless an improvement for the disabled user. The recoloring is done by inserting styles in the HTML document that overwrite the existing styles.

3 Swarm intelligence based optimization for solving the problem

For the minimization of F , we considered two swarm based optimization methods: Artificial Bee Colony and the API metaheuristic. In the following, we denote by $\mathcal{U}(X)$ and $\mathcal{R}(X \sim P)$ a random value in X which is uniformly distributed in the first case or distributed according to the probability distribution P in the second case. In the following, a recoloring is represented as an integer vector of dimension $3|\mathcal{C}|$. The search space is $\mathcal{S} = \llbracket 0 : 255 \rrbracket^{3|\mathcal{C}|}$. The best solution is memorized when F is computed and stored in s^* . We suppose that values are truncated to \mathcal{S} when it is appropriate.

3.1 Artificial Bee Colony

Bees inspired many optimization algorithms [8]. Artificial Bee Colony (ABC) [9] is one of the most popular. ABC principles are defined by the foraging behavior of onlooker bees, employed bees and scout bees. We denote by $\mathbb{S} = \{s_1, \dots, s_{\mathbb{S}}\}$ the food sources and by e_i the fail counter associated to s_i . e_{Max} is the maximum number of failure allowed for a food source before abandoning it. We denote by $\nu(x, y)$ the creation of a solution from the two food sources x and y . Let k be a random number in $\llbracket 1 : 3|\mathcal{C}| \rrbracket$ then we have, for all $i \neq k$, $\nu(x, y)_i = x_i$ and

```

/* Choose initial food sources */
for  $i = 1$  to  $|\mathbb{S}|$  do
|  $s_i \leftarrow \mathcal{U}(\mathcal{S})$ ;  $e_i \leftarrow 0$ 
while not done do
| /* Employed bees go out */
| for  $i = 1$  to  $|\mathbb{S}|$  do
| |  $v_i \leftarrow \nu(s_i, \mathcal{U}(\mathbb{S} - \{s_i\}))$ 
| | for  $i = 1$  to  $|\mathbb{S}|$  do
| | | if  $F(s_i) > F(v_i)$  then  $s_i \leftarrow v_i$ ;  $e_i = 0$  else  $e_i \leftarrow e_i + 1$ 
| | /* Onlooker bees exploit food sources */
| | for  $i = 1$  to  $|\mathbb{S}|$  do
| | |  $p_i = (1/(1 + F(s_i))) / \sum_{s \in \mathbb{S}} (1/(1 + F(s)))$ 
| | | for  $i = 1$  to  $\mathcal{N}_{onlooker}$  do
| | | |  $x_i \leftarrow \mathcal{R}(\llbracket 1 : |\mathbb{S}| \rrbracket \sim \mathcal{P})$ ;  $w_i \leftarrow \nu(s_{x_i}, \mathcal{U}(\mathbb{S} - \{s_{x_i}\}))$ 
| | | | for  $i = 1$  to  $\mathcal{N}_{onlooker}$  do
| | | | | if  $F(s_{x_i}) > F(w_i)$  then  $e_{x_i} = 0$ ;  $s_{x_i} \leftarrow w_i$  else  $e_{x_i} \leftarrow e_{x_i} + 1$ 
| | /* Employed bees abandon useless sources and become scout bees */
| |  $x = 1$ ;  $\mathcal{C} = \{i \in 1..|\mathbb{S}| \mid e_i \geq e_{\text{Max}}\}$ 
| | while  $x \leq \mathcal{N}_{scout}$  et  $\mathcal{C} \neq \emptyset$  do
| | |  $i = \arg \max_{j \in \mathcal{C}} e_j$ ;  $s_i \leftarrow \mathcal{U}(\mathcal{S})$ ;  $e_i \leftarrow 0$ ; Update  $\mathcal{C}$ ;  $x \leftarrow x + 1$ ;

```

Algorithm 1: The ABC algorithm for the textual color problem

$\nu(x, y)_k = x_k + \mathcal{U}(\llbracket -(x_k - y_k); (x_k - y_k) \rrbracket)$. Algorithm 1 proceeds in three steps. First, the employed bees go out of the colony toward their food sources and explore a new solution in its neighborhood. If the new solution is better than the food source, the food source is replaced and the fail counter is reset. Otherwise, the fail counter is increased. Second, the onlooker bees spread over the food sources according to their qualities $(1/(1 + F(s_i)))$ and explore a new solution in the neighborhood of the chosen food source. The food source and the fail counter are updated as for the employed bees. Finally, the food sources whose fail counter exceeds e_{Max} are abandoned and replaced by a new random food source. The process is repeated as needed. More detailed presentations of ABC can be found in the literature.

3.2 The API metaheuristic

The API metaheuristic [10] is inspired by the foraging strategy of primitive *Pachycondyla apicalis* ants [11, 12] which do not use pheromones. In this paper, we introduce only a shortened description of the algorithm. More details can be found in [10, 13]. We consider a colony of n ants. Each ant i has an associated hunting site denoted by $s_i \in \mathcal{S}$. If a hunting site is not defined, we have $s_i = \emptyset$. The behaviour of the algorithm is completely defined by two operators: $\mathcal{O}_{\text{init}}$ and $\mathcal{O}_{\text{explo}}$. $\mathcal{O}_{\text{init}}$ defines the initial position of the nest (\mathcal{N}) of the colony and, in this work, it is set to c^I . $\mathcal{O}_{\text{explo}}$ defines the foraging strategy of ants. $\mathcal{O}_{\text{explo}}(x, \mathcal{A})$ generates a solution y in the neighborhood of a solution x such that $\|x - y\|_{\text{max}} \leq \lfloor \mathcal{A} * 255 \rfloor$. \mathcal{A} is the neighborhood amplitude and takes its values in $[0 : 1]$. The operator is used to create hunting sites and to explore the neighborhood of

```

/* Choose initial nest position */
 $\mathcal{N} = c^I$ 
for  $i = 1$  to  $n$  do  $s_i \leftarrow \emptyset$  while not done do
    for  $i = 1$  to  $n$  do
        if  $s_i = \emptyset$  then
            /* Create a hunting site */
             $s_i \leftarrow \mathcal{O}_{\text{explo}}(\mathcal{N}, \mathcal{A}_i^{\text{site}})$ ;  $e_i \leftarrow 0$ 
        else
            /* Explore the neighborhood of a hunting site */
             $p \leftarrow \mathcal{O}_{\text{explo}}(s_i, \mathcal{A}_i^{\text{local}})$ 
            if  $F(p) < F(s_i)$  then
                 $s_i \leftarrow p$ ;  $e_i \leftarrow 0$ 
            else
                 $e_i \leftarrow e_i + 1$ 
                if  $e_i \geq e_{\text{Max}}$  then  $s_i \leftarrow \emptyset$ 
        /* Update nest if needed */
        every  $\mathcal{T}_{\text{move}}$  iterations do
             $\mathcal{N} \leftarrow s^*$ 
            for  $i = 1$  to  $n$  do  $s_i \leftarrow \emptyset$ 

```

Algorithm 2: The API metaheuristic for the textual color problem

hunting sites. The algorithm relies on two sets of amplitudes: $\{\mathcal{A}_i^{\text{site}}\}$ used for the hunting sites and $\{\mathcal{A}_i^{\text{local}}\}$ used for the exploration of hunting sites. In this work, we considered the classical definitions which are $\mathcal{A}_i^{\text{site}} = 1 + (0.01 - 1) \frac{i-n}{1-n}$ and $\mathcal{A}_i^{\text{local}} = \mathcal{A}_i^{\text{site}}/10$. We denote by e_i the counter for the number of failure of a hunting site s_i and e_{Max} the maximal number of failure allowed before a hunting is abandoned. Algorithm 2 proceeds in two steps. First, the ants go out of the nest. If the ant does not have a hunting site ($s_i = \emptyset$) then a new hunting site is chosen with $\mathcal{O}_{\text{explo}}(\mathcal{N}, \mathcal{A}_i^{\text{site}})$. If the ant already has a hunting site then it explores a new solution in the neighborhood of the hunting site using $\mathcal{O}_{\text{explo}}(s_i, \mathcal{A}_i^{\text{local}})$. When the new solution is better than the hunting site, the hunting site is replaced by the new solution and the fail counter is reset. Otherwise, the counter is increased and depending of its value, the hunting site is abandoned. When all ants have explored a new solution, the nest is tested. Every $\mathcal{T}_{\text{move}}$ iterations, the nest \mathcal{N} is moved to the best known solution and the hunting sites of the ants are cleared. The whole process is repeated as needed.

3.3 Hybridization with a line search

To try to speed up convergence of the algorithms, we hybridize them with a line search. Let $x \in \mathcal{S}$ and $v \in \llbracket -255 : 255 \rrbracket^{3|C|}$ a descent direction such that $F(x + v) < F(x)$ and $M = \|v\|_{\text{max}} > 0$. We define $(y_t)_{t \in \mathbb{N}^+}$ the sequence of vectors such that $y_t = x + v + (t/M)v$. By construction, we have $\|y_t - y_{t+1}\|_{\text{max}} = 1$. The line search consists in computing y_T such that $\forall t = 1..T-1$, $F(y_t) \geq F(y_{t+1})$ and $F(y_T) < F(y_{T+1})$. For the hybridization with ABC and API, the line search is

applied on any solution returned by $\nu(x, y)$ or $\mathcal{O}_{\text{explo}}(x, \mathcal{A})$ considering x as the initial vector and v as the difference between x and the initial returned value.

4 Experimental study and comparison

To compare the two methods, we consider different parameter settings and a set of 196 real web pages chosen randomly on Internet verifying $F(c^I) > 0$. The algorithms are run 50 times and the averaged “best fitness ever found” curves are computed for each pages. To alleviate scaling issues, the curves are normalized for every page such that $F(c^I)$ is mapped to 1 and the best found solution by any algorithm/parameter setting is mapped to 0. The averaged curve for an algorithm/parameter setting is obtained using the normalized curves on each page. The normalization formulas are given in [5]. For ABC, we considered $|\mathbb{S}| = \mathcal{N}_{\text{onlooker}} = \mathcal{N}_{\text{scout}} \in \{10, 20\}$ and $e_{\text{Max}} \in \{5, 10\}$. The initials colors c^I are included or not in the initial food sources and the usefulness of the line search is evaluated. From experiments, we noted that the inclusion of c^I in the initial food sources is very important and that the convergence is faster with 10 food sources than with 20. The use or not of the line search and the values of e_{Max} have very low influence on the results. Then, we consider the best ABC settings are $|\mathbb{S}| = \mathcal{N}_{\text{onlooker}} = \mathcal{N}_{\text{scout}} = 10$, $e_{\text{Max}} = 5$, c^I is included in the initial food sources and the line search is not used. For API, we considered setting the initial nest position to c^I or not, using the line search or not, setting $n \in \{10, 20\}$, $\mathcal{T}_{\text{move}} = 20$ and $e_{\text{Max}} \in \{5, 10\}$. From experiments, none of the parameter settings showed a significant influence on the behavior of the algorithm for the considered instances of the color problem. Arbitrarily, we consider the best API settings are obtained when c^I is the initial nest, the line search is not used, $n = 10$ and $e_{\text{Max}} = 5$ (similarly to ABC). The comparison of the best ABC and API settings showed that on most web pages, API converges faster than ABC. However, on some web pages, API fails to improve the initial solution while ABC achieves to improve it. Supposing a fair time sharing between the two algorithms, it is possible to run API and ABC in parallel in order to take the best of the two. To be fair, the same quantity of evaluated solutions (half) are used. The averaged curves on all pages are given by Fig. 1. This parallel approach is the most efficient. It demonstrates a fast convergence suitable for the interruption of the algorithm at any time and all web pages are improved (since the algorithm never failed). A more complete parameters study will be conducted in future work.

References

1. République Française: Loi n°2005-102 du 11 février 2005 pour l’égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées. JO n° 36 du 12 février 2005, p2353 (2005)
2. Colas, S., Monmarché, N., Burger, D., Mohamed, S.: A web site migration support tool to reach european accessibility standards. In: 9th European conference for the Advancement of Assistive Technology in Europe. Volume 20. (2007) 907–911

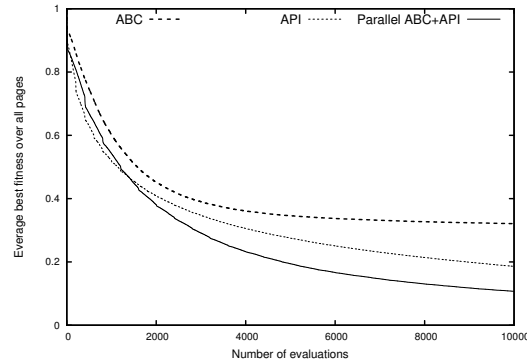


Fig. 1. Averaged curves on all pages for ABC, API and parallel ABC-API

3. Aupetit, S., Rouillé, V.: Annotation tool for the smart web accessibility platform. In: Proceedings of the 14th International Conference on Computers Helping People with Special Needs (ICCHP2014). (7-8 july 2014)
4. Mereuță, A., Aupetit, S., Slimane, M.: Improving web accessibility for dichromat users through contrast preservation. In: Computers Helping People with Special Needs. Volume 7382., Springer (2012) 363–370
5. Aupetit, S., Mereuță, A., Slimane, M.: Automatic color improvement of web pages with time limited operators. In: Computers Helping People with Special Needs. Volume 7382., Springer (2012) 355–362
6. Mereuta, A., Aupetit, S., Monmarché, N., Slimane, M.: Web page textual color contrast compensation for CVD users using optimization methods. Journal of Mathematical Modelling and Algorithms in Operations Research (October 2013)
7. Mereuta, A., Aupetit, S., Monmarché, N., Slimane, M.: An evolutionary approach to contrast compensation for dichromat users. In: Proceedings of Evolution Artificielle 2013, Lecture Notes in Computer Science, Springer (2013) to appear.
8. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artificial Intelligence Review (March 2012)
9. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty Computer Engineering Department, Kayseri/Türkiye (October 2005)
10. Monmarché, N.: Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation. Thèse de doctorat, Laboratoire d'Informatique de l'Université François Rabelais Tours (20 December 2000)
11. Fresneau, D.: Individual foraging and path fidelity in a ponerine ant. Insectes Sociaux, Paris **32**(2) (1985) 109–116
12. Fresneau, D.: Biologie et comportement social d'une fourmi ponérine néotropical (Pachycondyla apicalis). Thèse d'état, Université de Paris XIII, Laboratoire d'Ethologie Expérimentale et Comparée, France (1994)
13. Aupetit, S., Monmarché, N., Slimane, M.: Training hidden Markov models using the API ant algorithm. In: Artificial ants: from collective intelligence to real-life optimization and beyond. ISTE, Wiley (2010)

Parallel and Distributed Implementation Models for Bio-inspired Optimization Algorithms — Extended Abstract

Hongjian Wang, Jean-Charles Créput

IRTES-SeT, Université de Technologie de Belfort-Montbéliard, 90010 Belfort, France

Key words: Parallel and distributed computing, Metaheuristic optimization, Self-organizing map, Euclidean traveling salesman problem, Graphics processing unit

In the combinatorial optimization community, there exist a number of different bio-inspired optimization metaheuristics, such as genetic algorithms (GA), ant colony optimization (ACO) algorithm, artificial neural networks (ANN), and swarm intelligence based optimization (SIBO). Inspired by natural systems and designed to mimic certain phenomena or behaviors of biology, these algorithms aim at finding, as optimally as possible, approximate solutions to real-life difficult problems which are usually not able to be solved by exact approaches in reasonable computing time. Biologic systems are usually made up of populations of simple individuals, ants, birds or neurons, interacting locally with one another and with their environment. This trait should imply some potential for parallel and distributed implementation of the derived bio-inspired optimization algorithms. This makes sense for researchers to work on parallel and distributed computation models for the bio-inspired optimization algorithms. However, this implantation, from nature to practical parallel and distributed computational systems, is not as smooth as it looks like, owing to various restrictions of the latter coming from resource sharing and competition, communication and synchronization among computing nodes, system robustness requirement. Gains from different parallelism philosophies and implementation strategies may vary widely, and it is very tricky to come out with a consummate model. Trying to cast some interesting insights on this issue, this paper firstly contributes with a new taxonomy for various parallel and distributed implementation models of metaheuristic optimization, and then provides a parallel model of the Kohonen’s self-organizing map (SOM) ANN [1], as a representative of the “control distributed, data decomposition, memory centralized” category according to our taxonomy.

Generally, parallel computing speeds up computation by dividing the work load among a certain amount of processors. In the parallel computing community, two main sources of parallelism which are well accepted are *data parallelism* and *control parallelism*. Data parallelism refers to the execution of the same

operation or instruction on multiple large data subsets at the same time [2]. This is in contrast to control parallelism (or task parallelism, or function parallelism, or operation parallelism), which refers to the concurrent execution of different tasks allocated to different processors, possibly working on the “same” data and exchanging information [3]. Parallel computation based on these two parallelisms is particularly efficient when algorithms manipulate data structures that are strongly regular, such as matrices in matrix multiplications. Algorithms operating on irregular data structures or on data with strong dependencies among the different operations remain difficult to parallelize efficiently using only data or control parallelism. Metaheuristics generally belong to this category, and parallelizing them offers opportunities to find new ways to use parallel and distributed computational systems and to design parallel algorithms [4]. In our opinion, the traditional classification of parallelism looks not sufficient for the parallel metaheuristic optimization. One important point that should be emphasized concerns the allocation of processors according to the instance size of the problem. We think this point, specific to optimization, should be alighted in the taxonomies of metaheuristic parallel implementations, since it determines the maximum size of the input that should be solved in parallel computers and how the performances should grow according to the number of physical cores.

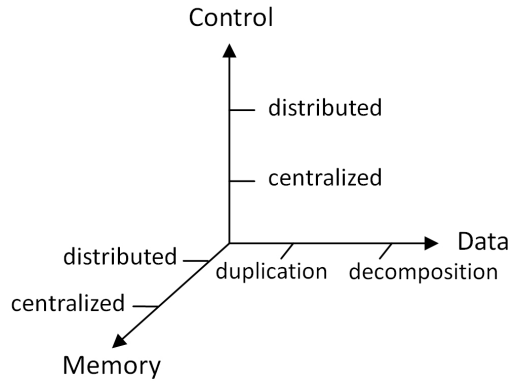


Fig. 1. Parallel and distributed computation model taxonomy based on control, data, and memory.

We propose a new taxonomy, as shown in Fig. 1, based on the three factors that every parallel metaheuristic implementation needs to consider: 1) control — is the control centralized or distributed? 2) data — is the data based on duplication or decomposition? 3) memory — is the memory centralized or distributed? In our opinion, one interesting model for SOM ANN should be attributed to the category of “control distributed, data decomposition, memory centralized”, in that, firstly, distributed control guarantees the model’s robustness, secondly,

data decomposition eases the burden of massive memory usage when dealing with large-scale problems, and thirdly, centralized memory reduces the communication costs among different processing units and allows easy implementation on Graphics Processing Unit (GPU) like systems.

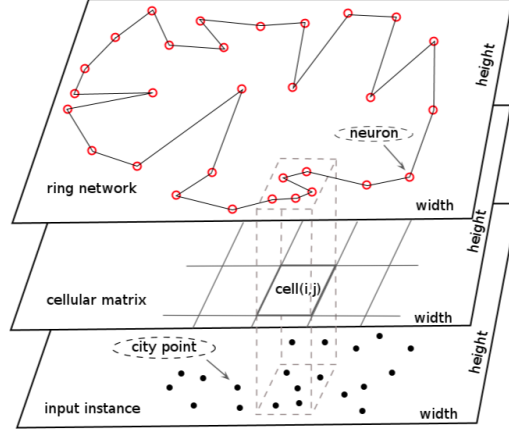


Fig. 2. Parallel cellular model: the input data density distribution, the cellular matrix and the neural network. To a given cell of the cellular matrix corresponds a constant part of the input data as well as a part of the neural network made up of SOM's topological grids / neurons.

In the parallel SOM model, that we propose to deal with the travelling salesman problem (TSP), we introduce a level of decomposition of the plane and input data. As illustrated in Fig. 2, three main data structures are used to implement the parallel model. Between the neural network and the input data, we add a uniform two-dimensional cellular matrix with linear relationship to the input size. Its role is to memorize the neurons in a distributed fashion and authorize many parallel closest point searches in the plane by a spiral search algorithm [5], and many parallel training procedures. Each uniformly sized cell in the cellular matrix is a basic training unit and will be handled by one parallel processing unit. Thus, the model proceeds from a cellular decomposition of the input data, in Euclidean space, such that each processing unit represents a constant and small part of data. Hence, according to the increase of parallel processing units in the future, the approach should be more and more competitive, while at the same time being able to deal with very large size inputs. This quintessential property holds because of the linear memory and processing units needed according to the input size. We discuss the relationships of this model to current parallel implementations of natural metaheuristics that are often based on data duplication rather than data decomposition.

References

1. Kohonen, T.: Self-organizing maps. Volume 30. Springer (2001)
2. Freitas, A.A., Lavington, S.H.: Data parallelism, control parallelism, and related issues. In: Mining Very Large Databases with Parallel Processing. Springer (2000) 71–78
3. Crainic, T.G., Toulouse, M.: Parallel meta-heuristics. In: Handbook of metaheuristics. Springer (2010) 497–541
4. Crainic, T.G., Toulouse, M.: Parallel strategies for meta-heuristics. Springer (2003)
5. Bentley, J.L., Weide, B.W., Yao, A.C.: Optimal expected-time algorithms for closest point problems. ACM Transactions on Mathematical Software (TOMS) **6** (1980) 563–580

Image post-segmentation correction using ant colony optimization

Ahmed Nasreddine Benaichouche, Hamouche Oulhadj, and Patrick Siarry

Laboratoire Images, Signaux et Systèmes Intelligents, LISSI (E.A. 3956)
122, rue Paul Armangot, 94400 Vitry sur Seine, France
`ahmed-nasreddine.benaichouche@univ-paris-est.fr`, `oulhadj@u-pec.fr`,
`siarry@u-pec.fr`

<http://www.lab.lissi.fr>

Abstract

Image segmentation is the process of partitioning a digital image into non-overlapped homogeneous regions with respect to some characteristics, i.e. pixels which present similar characteristics have to be grouped in the same cluster. Several image segmentation methods have been proposed in the literature, but unfortunately, none provides perfect results. With the aim of improving the segmentation results, we propose, in this work, a novel post-segmentation correction algorithm that can be applied after any segmentation method. Segmentation errors, often, occur at the boundaries between the detected regions. The first step of our method is the detection of the potentially misclassified pixels. The second step is the reclassification of these pixels basing on their local neighborhood statistics and the global quality of detected edges. Thus, we defined two objectives, a local objective for each pixel and a global objective for all potentially misclassified pixels. The local update of the pheromone of the ant colony system depends on the local objective and the global update depends on the global objective. The proposed method was tested on several noised synthetic images. The results show the success of the technique in improving the segmentation results.

Keywords: Image segmentation, post-segmentation, ant colony system

Multi-level parallelization for hybrid ACO

Omar Abdelkafi, Julien Lepagnot, and Lhassane Idoumghar

Université de Haute-Alsace (UHA)

LMIA (E.A. 3993)

4 rue des frères lumière, 68093 Mulhouse, France

{omar.abdelkafi, julien.lepagnot, lhassane.idoumghar}@uha.fr

Abstract. The Graphics-Processing-Unit (GPU) became one of the main platforms to design massively parallel metaheuristics. This advance is due to the highly parallel architecture of GPU and especially thanks to the publication of languages like CUDA. In this paper, we deal with a multi-level parallel hybrid Ant System (AS) to solve the Travelling Salesman Problem (TSP). This multi-level is represented by two parallel platforms. The first one is the GPU, this platform is used for the parallelization of tasks, data, solution and neighborhood-structure. The second platform is the MPI which is dedicated to the parallelization of programs. Our contribution is to use these two platforms to design a hybrid AS with a Local Search and a new heuristic.

Keywords: Parallel hybrid metaheuristics, TSP, GPU, MPI.

1 Introduction

Hybrid metaheuristics [1][2][3] are one of the most efficient classes of algorithms. The idea is to combine metaheuristics [4] and other techniques for optimization. With the combination of different techniques, these methods can require a longer computation time than others. This is one of the reasons that lead the community to propose parallel hybrid metaheuristics [5]. Another reason is the evolution of highly parallel architectures like the GPU. This evolution is due to the explosion of the industry of video games and his greedy demand for graphic power. Indeed, with the advent of CUDA, the use of GPU for non-graphic applications has become easier and hybrid metaheuristics have taken advantage of this evolution.

There are many levels of parallelization. For the Ant Colony Optimization (ACO) [6] applied to the TSP in the context of a single colony, the parallel execution of ants in the tour construction phase was initiated by Bullnheimer et al [7]. Also in this same context, in 2013, Cecilia et al [8] used the data parallelization in the update of pheromone to get the best performance from the GPU. In the context of multiple colonies, Stutzle [9] introduced the execution of multiple colonies in parallel with cooperation between colonies to improve the quality of solutions using the parallelization of programs. In CUDA programming, the execution on GPU is conducted by the kernel. It is a code called from the CPU (the host) and duplicated on GPU (the device) to run in a parallel way. The

kernel is executed in a *grid*, which is a set of *blocks* where every block is a set of *threads*.

In this work, we propose a hybrid ACO through one of the first variant of this method named the Ant System (AS) [10]. Our first contribution is to propose a new design for AS multi-colonies using GPU and MPI and the second contribution is to hybridize this method with a parallel local search (PLS) providing the intensification of the search and a new heuristic to improve results.

The rest of the paper is organized as follows. In section 2, we introduce the background needed for ACO and TSP to help the understanding of this proposition. We describe in section 3 the design of our multi-level parallel hybrid AS before we discuss the results of our experimentation in section 4. Finally, in section 5, we conclude the paper and we propose some perspective.

2 Background

The TSP is an NP-hard problem and one of the most studied combinatorial problems. It consists in finding the least-cost Hamiltonian circuit between a set of cities starting and ending with the same city. In general, TSP can be represented by a complete undirected graph $G = (V, E)$. The set $V = \{1, \dots, n\}$ is the vertex set, $E = \{(i, j) : i, j \in V, i < j\}$ is an edge set. c_{ij} is defined on E as the Euclidean distance between two vertices i and j .

Intuitively in the natural behaviors, the ants search the food randomly in the first tour construction. They move from one point to another until they find food. Once it is done, ants get back to the starting point. This corresponds to the initialization. In the search process, ants depose pheromone along the path they take. The quantity of pheromone is implemented by equation (1):

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij}^k \quad \forall (i, j) \in E \quad (1)$$

where $\Delta\tau_{ij}^k$ is the sum of pheromone which ant k deposits when it uses the edge between i and j . It depends on the length of the tour C^k constructed by the ant k ; $\Delta\tau_{ij}^k$ is defined in equation (2):

$$\Delta\tau_{ij}^k = \frac{1}{C^k} \quad (2)$$

Another characteristic of the pheromone in the natural behavior is the evaporation: the pheromone evaporates over time. This characteristic is implemented with a parameter $0 < \rho \leq 1$ in equation (3):

$$\tau_{ij} = (1 - \rho)\tau_{ij}, \quad \forall (i, j) \in E \quad (3)$$

The second step is the tour construction. In the natural behaviors, the ants follow the pheromone to find the best tour. To implement this concept, a probability is defined in equation (4) where $n_{ij} = \frac{1}{c_{ij}}$, α and β are parameters and N_i^k is feasible neighborhood. A complete survey on ACO can be found in [6].

$$\Delta p_{ij}^k = \frac{[\tau_{ij}]^\alpha [n_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [n_{il}]^\beta} \quad (4)$$

3 Design of the parallel hybrid ACO

The most straightforward way to design parallel AS or ACO in general is the parallelization of ants. This kind of parallelization is called the task parallelization and this is our first parallel level. The idea is very simple and used in most of the parallel ACO algorithms. Every ant is represented by a thread and every thread performs the tour construction in parallel with other ants. Inside the kernel, the ant chooses the next city to visit among the cities not selected yet and according to the probability computed by equation (4). The *CURAND library* allows the generation of a different random tour for every ant. The classical roulette wheel is used to select the next city to visit.

For the pheromone update part (see equation 1), using task parallelization can lead to concurrent access problems, i.e. if several ants update the pheromone of the same arc at the same time. The only solution in this case is to use atomic instructions but it decreases dramatically the performance. Hence, we are rather using data parallelism proposed by [8].

The level of data parallelization is used for the kernel of Update pheromone (see algorithm 1), the Evaporation pheromone (see algorithm 2) and the Update probability (see algorithm 3).

Algorithm 1 the Update pheromone kernel:

```

1: Input: Pants: the population of ants; fants: the fitness of ants; pheromone:
   the matrix of pheromone; cities: the size of the instance; ants: the size of the
   population;
2: Get the index of the thread idx; /*each idx represent one couple of cities*/
3: for i:=1 to ants do
4:   distance = fants[i];
5:   for j:=1 to cities do
6:     if the arc between i and j == idx then
7:       pheromone[idx]=pheromone[idx]+( $\frac{1}{distance}$ );
8:     end if
9:   end for
10: end for
```

Algorithm 2 the Evaporate pheromone kernel:

```

1: Input: pheromone: the matrix of pheromone;
2: Get the index of the thread idx; /*each idx represent one couple of cities*/
3: pheromone[idx]=(1-ρ) × pheromone[idx];
```

Algorithm 3 the Update probability kernel:

```

1: Input: pheromone: the matrix of pheromone; probabilities: the matrix of prob-
   abilities; cij: the matrix of distances; cities: the size of the instance;
2: Get the index of the thread idx; /*each idx represent one couple of cities*/
3: /*control if the cities of the couple are the same*/
4: if cij[idx]  $\neq$  0 then
5:   arc = (pheromone[idx]) $^{\alpha}$   $\times$  ( $\frac{1}{cij[idx]}$ ) $^{\beta}$ 
6:   all = 0
7:   position =  $\lfloor \frac{idx}{cities} \rfloor$  /*Get the position of the couple in the matrix*/
8:   /*when j=position, cij[(position  $\times$  cities)+j]=0*/
9:   for j  $\in$  {0, 1, ..., position - 1, position + 1, ..., cities} do
10:    all += (pheromone[(position  $\times$  cities) + j]) $^{\alpha}$   $\times$  ( $\frac{1}{cij[(position \times cities) + j]}$ ) $^{\beta}$ 
11:   end for
12:   probability[idx] =  $\frac{arc}{all}$ 
13: end if

```

Our idea to hybridize ACO is to use a Parallel Local Search and a new heuristics that we name smart ants. These algorithms are added to AS, but it can be used for all the variants of ACO. The PLS is applied to a group of ants after the Tour construction. It is a classical local search but the differences are the evaluation and generation of neighborhood executed in parallel with the GPU. It consists in representing every item of the solution by a thread, which leads to a parallel execution of neighbors generation. The thread generates and evaluates the neighbor of its item and searches the best possible switch. At the end of the parallel execution, the algorithm searches the best results of all the threads. This is the third level of parallelization.

The aim of the smart ant heuristic is to improve results. It executes as much iterations as the size of the instance without considering the start city which is static and unchangeable. The figure 1 shows a small example of the heuristic using 4 cities which mean 3 iterations and every vector represents an ant. In every iteration i we search the best ant inside the colony. For example in iteration 2 of the figure 1, the best ant is the third ant which have the index 2 because it starts from 0. All the ants follow the movement of ant_2 at the position 2 indicated by the arrow in the figure 1. The city in this position for ant_2 is city number 3. By consequence, ant_0 and ant_1 move their cities to get the city 3 in position 2. This is why we name it smart ants, because they have the intelligence to adjust their tour. All the ants perform this heuristic in parallel so we use the level of parallel tasks. As we can see with this heuristic, after a certain number of iterations, all the ants have the same tour. By consequence, this heuristic leads the search to stagnancy. To escape from this stagnancy, one improvement is added. The switch is not performed when the two cities to switch are adjacent (example in figure 1 the iteration 1 for ant_0).

The last step of our approach is to use MPI to execute our method on many GPU. This step introduces a new level of parallelization: the level of parallel programs. Actually, different colonies will be executed in parallel through many processes. For example, if we execute 3 processes, we will duplicate our algorithm

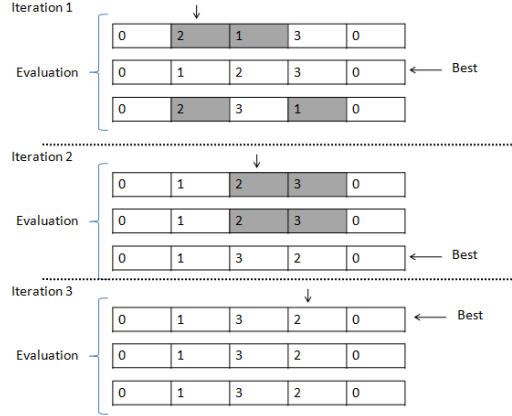


Fig. 1. smart ants

3 times. By consequence, 3 colonies will be executed in parallel. MPI gives to our design another advantage: we can exchange information between processes in order to improve the results. To exchange information, the algorithm regularly chooses the best solution found in one process and updates the pheromone of the matrix located in the next process using a ring topology. The data parallelization is not suitable this time. Since inside the solution every city is visited only once, a new level of parallelization between cities is applied which is the solution level parallelization. For all the couple of cities used in the tour, the pheromones of these couples are updated in parallel. The atomic operation is not needed because each couple appears only once in a tour.

4 Experimental results

4.1 Platform and tests

In our experimentation, we use a cluster of 12 graphic cards NVIDIA Geforce GTX680. The benchmark used is a set of well known instances from the TSPLIB [11] with a size between 51 and 150 cities. All the results are expressed as a percentage deviation from the optimum. All the optimal solutions can be found in the online benchmark library TSPLIB.

4.2 The smart ants heuristic

Table 1 shows the performance of the proposed algorithm with and without the smart ants (SA) heuristic for one colony. 25 tests are performed for every instance with 100 iterations. SA heuristic improves the average results of the 25 tests in the 5 instances.

Table 1. Evaluation of the smart ants heuristic

Instances	AVG with SA	AVG without SA
Eil51	3.13%	3.81%
Berlin52	2.50%	3.14%
Eil76	5.64%	6.35%
Pr76	4.85%	6.14%
KroA100	4.67%	5.26%

4.3 The Parallel multiple colonies using MPI

We use the cluster with 12 GPU. 10 tests for each instance are performed for 10 instances from TSPLIB. Table 2 reports the best results (MIN), the worst results (MAX), the average results (AVG) and the average time required for the 10 tests. The parameters used are $\alpha = 1$; $\beta = 2$; $\rho = 0,5$. 300 iterations are executed for each colony and every one of them contains 256 ants. 12 processes are executed one per machine in the cluster. The number of colonies executed in parallel is 12. Every 10 iterations the processes exchange their best solutions using a ring topology. 60% of the average results are between 0 and 3%. From the 10 instances, 9 average results are inferior to 5%.

Table 2. The multi-level parallel hybrid AS

Instances	MIN (%)	MAX (%)	AVG (%)	Time (s)
Eil51	0.99	3.02	1.98	10.57
Berlin52	0.03	2.33	1.07	14
St70	1.53	3.2	2.61	25.96
Eil76	2.83	5.04	4.21	28.02
Pr76	1.99	3.55	2.76	31.44
Rat99	3.53	7.8	6.19	35.7
KroA100	2.41	3.61	3.17	55.92
Bier127	1.25	2.58	1.87	87.72
Ch130	1.86	3.11	2.51	72.2
Ch150	2.84	3.75	3.42	76.8

The next experiment has the aim to see the behavior of the cluster when the objective function is evaluated equally between one GPU and 8 GPU. In this experiment, the same number of ants is used in the two cases. Table 3 presents the average results of 10 tests for 4 instances. AVG 1 is the average for the first case, AVG 2 is the average for the second case and ACC is the acceleration of the cluster compared to one GPU. With these conditions, the parallel design with the cluster improves the results and gives accelerations between 1.22 and 1.84 times compared to one GPU.

The final experiment is to compare our approach to other methods from the literature. In table 4, works from the literature are used for the evaluation. 4 approaches are selected. [12] is an ACO algorithm for TSP and [13][14][15] are other approaches to solve TSP for 5 instances. [*] is our approach and the results are the percentage deviation from the optimum. The Friedman test [16], performed on these 5 problems with $\alpha = 5\%$, shows that we can reject the null hypothesis, i.e. there is at least one algorithm whose performance is different from at least

Table 3. MPI accelerations

Instances	AVG 1(%)	AVG 2 (%)	ACC
Berlin52	2.72	1.04	×1.73
Pr76	5.4	3.08	×1.40
Bier127	2.53	2.18	×1.22
Ch150	4.43	3.91	×1.84

one of the other algorithms. To know which algorithms are different, we perform paired comparisons. The critical value is $C=3.67$. The paired comparisons (see Table 5) show that the results obtained by [*] are different from those obtained by the four other approaches. From the above analysis, we can see that our hybrid algorithm is better and outperforms the other four metaheuristics.

Table 4. Literature comparison

Instances	[*]	[12]	[13]	[14]	[15]
Eil51	1.98	7.98	2.89	2.69	3.43
Berlin52	1.07	7.38	7.01	5.18	5.81
Eil76	4.21	12.08	4.35	3.41	5.46
Bier127	1.87	15.32	3	2.2	3.41
Ch130	2.51	24.15	2.82	2.82	2.82

Table 5. Paired comparisons

Instances	[12]	[13]	[14]	[15]
[*]	19	10	4	12
[12]	-	9	15	7
[13]	-	-	6	2
[14]	-	-	-	8

5 Conclusion and perspectives

This work has two main objectives. The first one is to design a parallel ACO which can run in a cluster of GPU. The second objective is to improve the quality of solutions and to be as close as possible to the global optimum.

For the first objective, we use Five levels of parallelization. The first one is the parallelization of tasks performed by the GPU, which helps us to parallelize ants for the tour construction and the smart ant heuristic. The second level is the parallelization of data performed by GPU, which help us to update and evaporate the pheromones and to update the probabilities. The third level is the parallelization of the neighborhood structure performed also by GPU. This level is essentially used to parallelize the neighborhood inside the PLS. The fourth level is the solution level parallelization, performed by the GPU and used to update the pheromone when the best solution is exchanged between colonies. Finally, the last level is the parallelization of programs performed by MPI. It

allows us to parallelize different colonies and to diversify the search as much as possible. For the second objective we hybridize the AS: we use two techniques. The first one is to add the PLS for the intensification of the search. The second technique is to test a new heuristics named smart ant to improve results.

In our future works, we plan to apply the proposed algorithm to other combinatorial problems like the quadratic assignment problem. Another perspective is to reuse the same design for other swarm intelligence methods like the particle swarm optimization.

References

1. C. Blum, J. Puchinger, G.R. Raidl, A. Roli, Hybrid metaheuristics in combinatorial optimization: A survey, *Applied Soft Computing*, v.11 n.6, pp.4135-4151, (2011).
2. J. Lepagnot, L. Idoumghar, D. Fodorean: Hybrid Imperialist Competitive Algorithm with Simplex approach: Application to Electric Motor Design, 2013 IEEE International Conference on Systems Man and Cybernetics (SMC), Manchester UK, pp. 2454-2459, October (2013).
3. M.I. Aouad, L. Idoumghar, R. Schott, O. Zendra, Sequential and Distributed Hybrid GA-SA Algorithms for Energy Optimization in Embedded Systems, the IADIS International Conference Applied Computing 2010, pp. 167-174, (2010).
4. C. Blum, A. Roli, Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Surveys*, Vol.35, pp.268-308, (2003).
5. C. Cotta, E.G. Talbi, E. Alba, Parallel Hybrid Metaheuristics, in *Parallel Metaheuristics: A New Class of Algorithms*, John Wiley and Sons, (2005).
6. M. Dorigo, T. Stutzle, *Ant Colony Optimization*, Bradford Company, USA, (2004).
7. B. Bullnheimer, G. Kotsis, C. Strauss, Parallelization strategies for the ant system, in: *Applied Optimization*, vol. 24, pp. 87-100, (1997).
8. J.M. Cecilia, J.M. Garcia, A. Nisbet, M. Amos, M. Ujaldon, Enhancing data parallelism for Ant Colony Optimization on GPUs, *J. Parallel Distrib. Comput.* 73, pp. 42-51, (2013).
9. T. Stutzle, Parallelisation strategies for ant colony optimization, *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, PPSN V, Springer-Verlag, New York, (1998).
10. M. Dorigo, *Optimization: learning and natural algorithms*, Ph.D. Thesis, Politecnico di Milano, Italy, (1992).
11. G. Reinelt, TSPLIB - A Traveling Salesman Problem Library, *ORSA Journal on Computing*, Vol. 3 No. 4, pp. 376-384. Fall, (1991).
12. U. Chirico, A java framework for ant colony systems, Technical report, Siemens Informatica S.p.A, (2004).
13. E.M. Cochrane, J.E. Beasley, The co-adaptive neural network approach to the Euclidean traveling salesman problem, *Neural Networks*, 16(10), pp. 1499-1525. (2003).
14. T. A. S. Masutti, L. N. D. Castro, A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem, *Information Sciences*, 179(10), pp. 1454-1468. (2009).
15. S.Somhom, A.Modares, T.Enkawa, A self-organizing model for the traveling salesman problem, *Journal of the Operational Research Society*, pp. 919-928, (1997).
16. L. Idoumghar, N. Chérin, P.Siarry, R. Roche, A.Miraoui, Hybrid ICA-PSO algorithm for continuous optimization, *Applied Mathematics and Computation*, Applied Mathematics and Computation 219, pp. 11149-11170, (2013).

Swarm Intelligence-based BPEL Partitioning Process for distributed orchestration

Takwa Mohsni¹ and Zaki Brahmi²

¹Department of Informatics, Jandouba University, Tunisia

²RIADI-GD Laboratory, National School of Computer Sciences, Manouba University,
Tunisia.

Emails: takwa.mohsnii, zakibrahmi@gmail.com

Abstract. *Business Process Execution Language*(BPEL), the standard technology to compose web services, is scheduled to execute in a centralized mode. Therefore, the centralized execution of BPEL is expensive in terms of response time and scalability. In counterParty, the decentralized execution of BPEL process offers performance improvements in terms of increased throughput and scalability and lower response time. The decentralized execution of BPEL process is done by partitioning a BPEL process into a set of BPEL sub-processes; each of them is committed to perform a part of the BPEL process. All the BPEL sub-processes must be coordinated to perform the overall Process execution. The BPEL partitioning program is an optimization problem with a high-dimensional search space.

In this paper, we propose a novel contribution based on swarm intelligence optimization, basing on its capacity to build a simple artificial societies of insects who can collectively provide a complex task, to solve the BPEL partitioning problem and optimizing its response time delay and throughput.

Keywords: Decentralized orchestration, BPEL Partitioning, Swarm Intelligence Optimization

1 Introduction

Business Process Execution Language-BPEL, is the industry standard used to express Web Services orchestrations. A BPEL process defines how multiple service interactions between partners can be coordinated internally in order to achieve a business goal (orchestration) [1][2]. Once its specification has been developed, the BPEL process is deployed in centralized server, which controls the execution flow and coordinates the interactions with partner services. The centralized management of BPEL process have many problems, including poor performance, impaired reliability, limited scalability, and restricted flexibility [3].

Decentralized orchestration offers performance improvements in terms of increased throughput and scalability and lower response time [3]. Decentralized orchestration can be done by different style of decentralization such as BPEL partitioning and message passing style. The BPEL program partitioning consist to divide BPEL program into a set of sub-programs. These sub-programs are executed on different servers and coordinate to execute the over program.

In BPEL partitioning problem, BPEL process is divided at two sets activities: fixed activities and mobile activities. The aim is to find the optimal allocation of mobiles activities with the fixed activities, in order to maximize the response delay and to minimize the throughput.

The BPEL partitioning program is an optimization problem with a high-dimensional search space. The classical BPEL partitioning algorithms such as Genetic algorithm [4], Merge-by-def-Use [5] and Pooling-and-Greedy Merge [5], do not provide a suitable solution because the search space increases exponentially with the problem size. Over the last decades, there has been a growing interest in optimization algorithms inspired by the behaviors of natural phenomena [6], [7], [8] and [9]. It is shown by many researchers that these optimization algorithms are well suited to solve complex computational problems such as task allocation [10][11], data analysis [12], robotics systems [13] and image processing [14].

In this paper, we propose a novel approach based on division of labour in ants colonies [15] as a solution for BPEL partitioning problem. The model of division of labour in ants colonies, assumed that there is N tasks need to be perform by M workers. *Stimuli*-associated to each task used to stimulate the workers. And, in counter party, each worker reacts with the sent stimuli by an *internal response*. The workers involved in the execution of the set of tasks with different probabilities. Fundamentally, we consider the BPEL partitioning program as division of labour problem. We assume that each fixed activity involved in the execution of the set of mobile activities with probability. This probability is calculated by stimuli-associated to each fixed activity and an internal response-associated to each mobile activity.

The remainder of the paper is organized as follows: Section 2 gives related previous works on BPEL partitioning problem for decentralized execution. In section 3, we formulate the research problem. After that, we present our new swarm algorithm for partitioning BPEL program in section 4 and the experimentation results in section 5. Finally, we conclude our research in section 6.

2 Related Works

The BPEL Partitioning problem was first introduced by Nanda et al.,2004 [5]. They presented two heuristics algorithms to partition a BPEL process presented as Program Dependence Graph (PDG). Heuristic partitioning algorithms named, respectively, *Merge-by-Def-Use*(MDU) and *Pooling-and-Greedy Merge* (PGM). The aim of the MDU partitioning algorithm is to determine the best partitions at which each portable task must be executed in order to optimize the throughput of the decentralized program. However, due to large computation time to MDU algorithm, the authors are choose to apply the PGM heuristic to solve this problem - (1) - the *greedy-merge* is a refinement to minimize the data on the network and - (2) - the *pooling* heuristic tries to minimize the total number of messages.

Another approach based on genetic algorithm [4] is proposed to partition BPEL process presented as Program Dependence Graph. In order to use the genetic algorithm, this runs on 4 steps; *Genetic encoding* - *Initial population generation* - *Genetic operators* and *Fitness Function*. The initial population generation consists to generate a set of genomes randomly ordered. The genome is encoded by an array of integers of length equal to the number of mobile nodes in the BPEL process. Each gene represents a mobile node and has value ranging from 0 to m , where m is the number of fixed node in BPEL process. Authors are used two genetic operators (crossover operator and one mutation operator) and a fitness function to explore the search space. A local optimizer is applied to each feasible individual in the population, to explore the neighboring solutions of the individual for possible improvement.

3 BPEL Partitioning Problem Statement

A description of the BPEL partitioning problem was firstly introduced by [5] as described below:

1. The BPEL program is represented by a set of activities, that can be classified into *Fixed activities* and *Mobile activities*. Fixed activity can be "receive", "reply" or "invoke" activity for the reason that must executed at the particular server. Other activities are mobile activities, including activities "switch", "assign", "if", and so on. The mobile activities can be executed at any server.
2. The BPEL activities present certain links between them. These links are described by two types of dependencies; (1) *Data dependencies*: present between two activities in the case where a variable which is defined in the first of them is referenced in the second. (2) *Control dependencies*: appears between two activities with the same control state.
3. Each Fixed activity must be assigned to separate sub-process. Thus, a sub-process has exactly one fixed activity and zero or more mobile activity.

Formally, we can define the PPBPEL problem of partitioning BPEL process as follows: $PPBPEL = \langle E, \theta, D, C, SPBPEL, \beta \rangle$ where :

- E a set of f fixed activities : $E = \{e_1, \dots, e_f\}$.
- θ a set of p mobile activities : $\theta = \{\theta_1, \dots, \theta_p\}$.
- D a set of data dependencies between *fixed activities* and *mobile activities* :
 $D = \{ \langle di, dj \rangle \mid di \text{ uses the data generated by } dj, \text{ where } di, dj \in E \cap \theta \}$.
- C a set of control dependencies between two activities BPEL :
 $C = \{ \langle ci, cj, k \rangle \mid \text{there exists a control dependency of type } k \text{ from } ci \text{ to } cj, \text{ where } ci, cj \in E \cup \theta \text{ and } k \in \{ \text{true}, \text{false} \} \}$.
- $SPBPEL$ a set of m sub-process ($\in (E \cup \theta)$) : $SPBPEL = \{ SP_1, SP_2, \dots, SP_m \}$, where :
 The set of sub-process SPBEPL must preserve the following requirements:
 1. $SP_i \cap SP_j = \emptyset$, if $i \neq j$ where $1 \leq i, j \leq f$,
 2. $SP_1 \cup SP_2 \cup \dots \cup SP_m = E \cup \theta$, and
 3. $SP_i \subseteq \{ e_1 \text{ or } e_2 \text{ or } \dots \text{ or } e_f \}$
- β the objective function to give the throughput of the given BPEL process under partitioning (equation 1):

$$\beta(BPEL)_{underPartitioning} = \min(S(SP_1), \dots, S(SP_n)) \quad (1)$$

with:

- n : is the number of communicating servers, each of which implements a portion of the overall service and,
- $S(SP_1)$: is the throughput delivered by each individual server.

4 Swarm intelligence-based approach

A swarm is a large number of homogeneous simple agents interacting locally among themselves and their environment, with no center to allow a global interesting behavior to emerge [16]. Swarm-based algorithms have recently emerged as a family of algorithms inspired from nature, that are capable of producing low, cost, fast and robust solutions to several complex problems such as Ant Colony Optimization [15], Particle Swarm Optimization [6], Artificial Bee Colony [7], Cat Swarm Optimization [8] and Artificial Immune System [9]. In this paper, we address to the model of division of labour in ants colonies [15].

4.1 Division of Labour in ants colonies

A Model of division of labour in ants colonies, based on fixed response thresholds was introduced in [15]. Authors assumed that m tasks need to be performed. These Tasks, denoted by j , are associated with stimuli or demands. Let s_j denote the intensity of task j -associated stimuli. And, there are N workers, denoted by i , with response thresholds θ_{ij} ($1 \leq i \leq N$ and $1 \leq j \leq m$) with respect to task j -associated stimuli. Therefore, individual i engages in task j with the following probability:

$$T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2} \quad (2)$$

Accordingly, each worker responds to a given stimuli when stimuli intensity exceeds the worker's threshold. The model (Bonabeau et al. 1996) can explain how flexibility at the colony level results from the workers' behavioral flexibility.

4.2 BPEL Partitioning Process based on division of labour in ants colonies

Inspired by the recruitment process among ants colonies, we assume that a set of mobile activities θ can be recruiter (performed) by a set of Fixed activities E . The question remains:

How the set of mobile activities were dived between the set of fixed activities responding to the local information of each activity?

According to the theoretical model of response threshold described in [15], we assume that each fixed activity $e_i \in E$ has a single Tendency value to perform a mobile activity $\theta_j \in \theta$. The Tendency reflects the capability of fixed activity e_i to perform a mobile activity θ_j . The fixed activities produce a *stimulus* for mobile activities as shown in Fig(1) that, because of their internal *response thresholds*, have different tendencies to recruit the mobile activities.

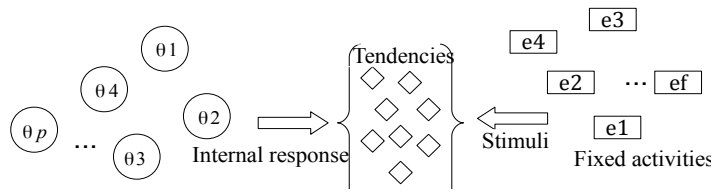


Fig. 1. Swarm system for BPEL process

Calculating Tendency The tendency $T_{\tau_{\theta_j}}(s_{e_i}) \rightarrow [0,1]$ of fixed activity e_i to recruit a mobile activity θ_j is determined by the stimulus and the internal response thresholds as shown in equation(3):

$$T_{\tau_{\theta_j}}(s_{e_i}) = \frac{s_{e_i}^2}{s_{e_i}^2 + \tau_{\theta_j}^2} \quad (3)$$

where :

- s_{e_i} Stimulus associated with fixed activity e_i to mobile activity θ_j .
- τ_{θ_j} Internal threshold for mobile activity θ_j to fixed activity e_i .

In our proposed approach, the stimulus s_{e_i} is determined by the capacity of server S_i that contains the fixed activity e_i . The stimulus decreases proportionally to the server capacity that's reduced each time when the server perform a mobile activity. Thus server with large capacity for a mobile activity θ_j has higher tendency to perform this mobile activity. The capacity of a server S_i is determined by its resources. Each server S_i has a limited number of resources denoted by $S_i.res$. The process of recruitment must respect all server resources limitation that's measured by the sum of the resource of the local fixed activity $e_i \in S_i$ denoted by $Res(S_i, e_i)$ and the resource needed to perform θ_j denoted by $Res(S_i, \theta_j)$. The $s_{e_i}(t)$ local stimulus associated with fixed activity e_i is calculated at each time t as (equation 4):

$$s_{e_i}(t) = S_i.res^t - \sum_{\theta_j^t \in \theta^t} Res(S_i^t, \theta_j^t) + Res(S_i^t, e_i^t) \quad (4)$$

Therefore, each mobile activity $\theta_j \in \theta$ has an internal response threshold τ_{θ_j} , is an internal variable, to respond to the stimulus $s_{e_i}(t)$ sent by the fixed activity e_i at time t . The internal response threshold τ_{θ_j} decreases proportionally to the fixed activity stimulus. It is given by equation(5):

$$\tau_{\theta_j}(t) = 1 - s_{e_i}(t) \quad (5)$$

Dependency Constraints The BPEL program activities are characterized by the existence of control dependencies and data dependencies among them, as noted in section 3. An effective partitioning plan must guarantee (respect) this dependency constraints. Responding to this problem, we assume that each mobile activity $\theta_j \in \theta$ must be recruited by a fixed activity $e_i \in E$ possessing data dependency $D(e_i, \theta_j)$ or control dependency $C(e_i, \theta_j, k)$ with it, where it is considered as the most capable to perform the current mobile activity. The fixed activities that don't have any dependency with the mobile activity θ_j , its Tendency to perform the current mobile activity must be reduced to zero (equation 6).

$$T_{\tau_{\theta_j}}(s_{e_i}) = \begin{cases} \frac{s_{e_i}^2}{s_{e_i}^2 + \tau_{\theta_j}^2} & \text{if } \exists D(e_i, \theta_j) \text{ OR } \exists C(e_i, \theta_j, k), k \in \{true, false\} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Optimal Tendency The goal of partitioning BPEL process is to find optimal partitions that maximize the BPEL process reward. However, a partitioning BPEL program algorithm must guarantee that each mobile activity belongs to the partition maximizing its performance. In other words, the mobile activity will create a partition with the fixed activity that has the maximum Tendency value denoted $T.max$ given by :

$$T.max = \max(L.Tendencies) \quad (7)$$

where :

- max : is the function that return the maximum Tendency value in the List $L.Tendencies$.
- $L.Tendencies$ represents the Tendencies values list of the fixed activities, where the Table length is proportional to the number of fixed activities and each case of the table contains the Tendency value of current fixed activity, corresponds to the case, to perform the mobile activity θ_j .

Algorithm Description Our Swarm intelligence-based algorithm is an approximate algorithm that uses to solve Partitioning BPEL Program Problem. The aim of algorithm is to allow mobile activity to decide individually with which fixed activity to be merges, in a simple and efficient way, minimizing communication costs and maximizing the throughput. Algorithm given below (Algorithm 1) present the details of our approach. The algorithm is executed by each mobile activity θ_j as decentralized fashion.

Algorithm 1 Swarm Intelligence-based algorithm for mobile activity θ_j

Input: $E = \emptyset$ //set of Fixed Activities
 $\theta = \emptyset$ //set of Mobile Activities
 $SPBPEL = \emptyset$ //Sub-Processes BPEL
 $BPEL$ //file containing the BPEL processes
Output: $SPBPEL = \{ SP_1, SP_2, \dots, SP_p \}$ // p Sub-Process BPEL
Begin
1: $(E, \theta) \leftarrow \text{Parsing}(BPEL)$ //Identify the set of fixed activities E and the set of mobile activities θ .
2: **for** each fixed activity $e_i \in E$ **do**
3: **if** (Data-DP(e_i, θ_j) Or Control-DP(e_i, θ_j)) **then**
4: //If θ_j has data dependency or control dependency with e_i . The Tendency (e_i, θ_j) is
5: $T_{\tau_{\theta_j}}(s_{e_i}) \leftarrow \frac{s_{e_i}^2}{s_{e_i}^2 + \tau_{\theta_j}^2}$
6: **else** //If θ_j not has any dependency with e_i . The Tendency (e_i, θ_j) reduced to zero
7: $T_{\tau_{\theta_j}}(s_{e_i}) \leftarrow 0$
8: **end if**
9: $L.Tendencies.addElement(Tendency(e_i, \theta_j))$
10: //When the Tendencies values are calculated, the list $L.Tendencies$ is created that contains the set of Tendencies values as indicated in section 4.2.
11: **end for**
12: $T.max \leftarrow \text{MaxTendencyValue}(L.Tendencies)$ //Seek the maximum Tendency value $T.max$ of the List of Tendencies values
13: picks fixed activity e_i which has $T.max$ //Seek the fixed activity that has the maximum Tendency value $T.max$
14: $SPBPEL.add(SP_i(e_i, \theta_j))$ //merges with the fixed activity which has $T.max$.
End

5 Experimental results

In this section, we test our Swarm Intelligence-based algorithm performance. To evaluate the results generated by our algorithm (SI), we compared it to the results generated by the genetic algorithm-based approach (GA)[4]. We implemented the two algorithms in eclipse 3.4.1 with Object-oriented language Java and executed on a desktop computer with Dual-core 2.30 GHz and 2 GB RAM.

The computation time and efficiency of the algorithms depend on the size and the complexity of the BPEL process problem [4]. The size of the problem is dependent on two parameters, the number of fixed activities in the BPEL process and the number of mobile activities in the BPEL process. The complexity of the problem largely depends on the levels of control dependencies constraints in the BPEL process. We used our algorithm (SI) based on swarm intelligence and algorithm (GA) of [4] based on the

genetic algorithm to solve each problem tests and we have recorded the computation time of each algorithm .

The first set of test problems included 6 test problems with different number of fixed and mobile activities. The results are shown in Fig(2). The Compute Time of GA is very large compared to the Swarm Intelligence-based algorithm(SI). We can see that the computation time of the GA increases largely as the problem size increases. In counter party, the computation time of our algorithm (SI) increases slowly as the problem size increases. This is because the computation time of the GA depends not only on the number of activities like our algorithm, but also on the corresponding partitioning topology of the solutions produced during evolving steps.

The second set of test problems included 4 test problems with different levels of control dependencies constraints as shown in Fig(3). The compute time of two algorithms increases as the level of complexities increases. The time to compute the GA is also large compared the time to compute our algorithm.

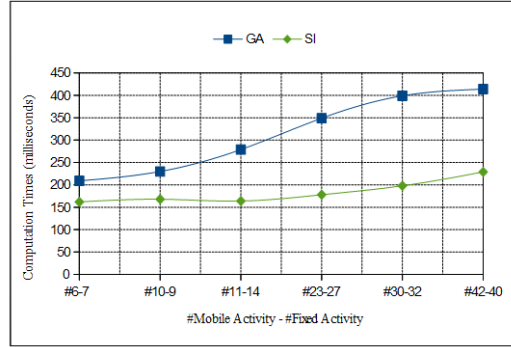


Fig. 2. Comparisons of the computation time of our Swarm Intelligence-based algorithm (SI) versus Genetic Algorithm (GA) for resultant partitioning of different BPEL Program sizes(in milliseconds)

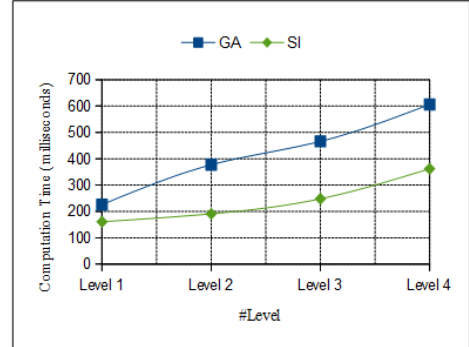


Fig. 3. Comparisons of the computation time of our Swarm Intelligence-based algorithm (SI) versus Genetic Algorithm (GA) for resultant partitioning of different complexities Levels(in milliseconds)

According to the results of the experiments above, we can see that the computation time of our algorithm based on swarm intelligence (SI) increases slowly when the problem size increases. However, the scalability of the genetic algorithm (GA) is poor. Its computation time seems to increase Largely with the problem size. However, the computation time also increased too rapidly in relation to the problem size. Therefore, we can conclude that our algorithm (SI) based on swarm intelligence has a high capacity to adapt to the increase in size and in the problem complexity. The results given presents a great performance of our algorithm via the size changes and the increase in complexity compared to the results given by the algorithm of [5] that is very sensitive to problem changes.

6 Conclusion

In this paper, we introduced our research efforts to address web services orchestration issues in decentralized and dynamic fashion. The approach introduced there- Swarm

BPEL- deals the problem of partitioning BPEL program for decentralized execution based on the swarm intelligence. This algorithm solves the Partitioning BPEL process an approximated and distributed fashion. As research perspectives, we propose to improve the performance of our algorithm by adding a priority constraint in the recruitment process. Another research direction is to envision the algorithm to merge the fixed nodes when multiple fixed nodes are placed in the same partition.

References

1. F. ABOUZAIID AND J. MULINS – "A Calculus for Generation, Verification and Refinement of BPEL Specifications", *Proc. of the 3rd International Workshop on Automated Specification and Verification of Web Systems, Electronic Notes in Theoretical Computer Science (2008)*, p. 43–65.
2. D. HABICH, S. RICHLY, M. GRASSELLT, S. PREISSLER, W. LEHNER AND A. MAIER – "BPEL - Data-Aware Extension of BPEL to Support Data-Intensive Service Applications", *In. Whitestein Series in software Agent Technologies and Automatic (2008)*, p. 111–128.
3. M. ILAHI, Z. BRAHMI AND M.M. GAMMOUDI – "Enhancing Decentralized MAS-Based Framework for Composite Web Services Orchestration and Exception Handling by Means of Mobile Agents Technology", *Proc. of AMT'09, Beijing-China, Vol. 5820/2009, Springer Berlin / Heidelberg (2009)*, p. 347–356.
4. AI. LIFENG, T. MOALIN AND F. COLIN – "Partitioning composite web services for decentralized execution using a genetic algorithm", *In. Future Generation Computer Systems. Elsevier Sciences Publisher (August 2011)*, Vol.27, p. 157–172.
5. M. G. NANDA, S. CHANDRA AND V. SARKAR – "Decentralizing Execution of Composite Web Services", *Proceedings of the 19th conference on Object-oriented programming, systems, languages, and applications OOPSLA '04(October 2004)*, p. 170–187.
6. R. EBERHAT AND J. KENNEDY – "A New Optimizer Using Particle Swarm Theory", *Proceedings of Sixth International Symposium on Micro Machine and Human Science (1995)*, p. 39–43.
7. D. KARABOGA – "An Idea Based On Honey Bee Swarm For Numerical Optimization", Technical report-TR06, Erciyes University. Engineering Faculty. Computer Engineering Department (October 2005).
8. S-C. CHU, P-W. TSAI AND J-S. PAN – "Cat Swarm Optimization", *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence. Lecture Note in Computer Science (2006)*, p. 854–858.
9. M. BAKHOUYA AND J. GABER – "An Immune Inspired-based Optimization Algorithm: Application to the Traveling Salesman Problem", *In. Advanced Modeling and Optimization (2007)*, p. 105–116.
10. F. DOS SANTOS AND L.C. BAZZAN – "An Ant Based Algorithm for Task Allocation in Large-Scale and Dynamic Multiagent Scenarios", *Proceedings of the 11th Annual conference on Genetic and evolutionary computation GECCO '09 (2009)*, p. 73–80.
11. P. SCERRI, A. FARINELLI, S. OKAMOTO AND M. TAMBE – "Allocating Tasks in Extreme Teams", *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems AAMAS '05 (2005)*, p. 727–734.
12. Z. LI, B. DING, J. HAN AND R. KAYS – "Swarm: Mining Relaxed Temporal Moving Object Clusters", *Proceedings of VLDB Endowment (September 2010)*, Vol.3, p. 723–734.
13. G. BENI AND J. WANG – "Swarm Intelligence in Cellular Robotic Systems", *Proceedings of NATO Advanced Workshop on Robots and Biological Systems (1989)*, p. 703–712.
14. S. SAATCHI AND C.C. HUNG – "Swarm Intelligence and Image Segmentation", *Swarm Intelligence and Image Segmentation, Swarm Intelligence, Focus on Ant and Particle Swarm Optimization, Felix T.S.Chan and Manoj Kumar Tiwari (Ed.), ISBN: 978-3-902613-09-7 (2007)*.
15. G. THERAULAZ, E. BONABEAU AND J. L. DENEUBOURG – "Response Threshold Reinforcement and Division of Labour in Insect Societies", *Proceedings of Royal Society. Biological Sciences (February 1998)*, Vol. 265, p. 327–332.

Particle Swarm Optimization for the Multi-Level Lot Sizing Problem

David Lemoine

Ecole des Mines de Nantes, IRCCyN UMR-CNRS 6597, La chantrerie, 4 rue Alfred Kastler,
BP 20722, F-44307 Nantes Cedex 3, France
david.lemoine@mines-nantes.fr

Abstract. The aim of tactical planning is to determine the quantities to be produced per period, in order to fulfill the customers' demands. To deal with this challenge, many optimization models called "Lot Sizing models" have been developed. Among them, a specific one, the "Multi Level Lot Sizing Problem" (MLLP), has been elaborated without taking into account the production capacities of the manufacturing system, according to the MRP process. The latter tries to determine a production plan for end-items as well as components by seeking to minimize logistics costs (setup and holding costs) and is known to be NP-Hard. In this paper, we proposed to use a Discrete PSO Scheme to pilot a level by level planning method by using a cost modification process. Unlike the DPSO elaborated by (Deroussi *et al.*, 2009) for the same problem, the proposed method is based on a respect of the cost's property of Wagner-Whitin and the philosophy of the costs' modifications methods. Consequently, this new method outperforms (Deroussi *et al.*, 2009) one, in terms of rapidity and efficiency on the literature benchmarks.

Keywords: Particle Swarm Optimization, Tactical Planning, Multi-Level Lot-sizing Problem, MLLP, Costs' modification, Wagner-Whitin Costs

Swarm projects: beyond the metaphor

Pierre Parrend^{1,2}, Pierre Masai, Cecilia Zanni-Merk¹, Pierre Collet¹

¹ ICube Laboratory, Université de Strasbourg, France
pierre.collet@unistra.fr, cecilia.zanni-merk@unistra.fr

² ECAM Strasbourg-Europe, Schiltigheim, France
pierre.parrend@ecam-strasbourg.eu

Keywords: swarm intelligence, agile, SCRUM Framework, stygmergy, emergent cognition

Extended Abstract

Swarming has become a management tool for letting individuals cooperate in order to generate emergent solutions to difficult issues in organizations. The idea behind this paper is to get people to apply and focus their intelligence on the submission of a proposal in a swarming way, so that the global obtained result is of a better quality than the sum of each individual submission of a partial solution to the problem.

The problem is that in order to get an emergent behaviour, it is necessary to come up with management practices which are capable to exploit the mechanisms of swarm intelligence at the level of the organization. Whereas we will show that swarming behaviour build the core of agile concepts, the lack of awareness of these mechanisms prevent teams and organizations from leveraging swarm dynamics.

We focus on an emergent process, namely agile software development, to propose a model of information generation and exchange between individuals of an organisation that would promote the emergence of a swarming phenomenon. This paper will bring insight to the dynamics of agile projects, and will provide an example of a swarming behaviour based on complex, though predictable, entities.

Agile software development is a generic naming for instances of development processes complying with the Agile Manifesto³ (2001). We focus on the SCRUM Framework, but other approaches such as Extreme Programming, or Crystal clear have been defined. In Agile projects, entities communicate through stigmergy in a constant effort toward transparency in technical as well as organizational and interpersonal issues. The cognitive model of an Agile development team has striking parallels with swarm cognitive models. The inspection and adaptation feedback loops of SCRUM in particular create emergent cognitive mechanisms capable of transforming individual actions in an organization's movement, and in aligning processes of individuals with the requirements and dynamics of the organization.

³ <http://agilemanifesto.org/>

The proposed model is challenged by several issues: How do the scaling of swarms and the scaling of Agile projects relate? How can one evaluate the impact of the careful implementation of swarm properties of Agile processes on the success of development projects? Is it possible to simulate an Agile team in order to validate its behaviour as a complex system, as what has been done in artificial life or swarming behaviour?

The analysis presented in this paper is illustrated by a controlled experiment involving 50 computer science students at the bachelor level, in the context of a 4 month development project involving 8 parallel teams working on the same software project.

Fuzzy Logic Control Optimized by Artificial Immune System for Building Thermal Condition

Jiawei Zhu¹, Fabrice Lauri¹, Abderrafaa Koukam¹, and Vincent Hilaire¹

IRTES-SET, UTBM, 90010 Belfort cedex, France

(`jiawei.zhu`, `fabrice.lauri`, `abder.koukam`, `vincent.hilaire`)@utbm.fr

Abstract. With the fast development of information technology and increasingly prominent environmental problems, building comfort and energy management become the major tasks for an intelligent residential building system. According to statistical studies, people spend 80% of their lives in buildings. Hence it is not surprising that they constantly seek to improve comfort in their living spaces. This paper presents a fuzzy logic controller optimized by an artificial immune system algorithm aimed at maintaining the thermal comfort while reducing energy consumption. The experimental results show the advantages of our system compared with the widely used baseline: On/Off control approach.

Keywords: energy, fuzzy system, artificial immune system, optimization

1 Introduction

According to statistical studies, people spend 80% of their lives in buildings. This explains why occupants constantly seek to improve comfort in their living spaces. In addition, environmental issues have drawn more and more attention. How to manage energy in a proper way to improve energy efficiency and reduce pollution is a subject of uttermost importance. Meanwhile, the popularization of the concept of home office makes the productivity in residential buildings economically significant.

Among all indoor comfort factors, thermal comfort attracts our special attention. According to [1], thermal comfort is the condition of mind which expresses satisfaction with the thermal environment. This definition leaves open what is meant by *condition of mind* or *satisfaction*, which implies that the judgement of comfort is a cognitive process involving many inputs including physical, physiological, psychological and other processes. Despite not being the only affecting factor, indoor temperature has physically major influence on occupants' feeling comfort. In real world, the operative temperature intervals vary with building location and type. ISO-7730 suggests temperature ranges in different types of buildings and different environmental conditions. For example, for residential buildings of category B in summer, the suggested temperature range is from 23.0°C to 26.0°C , while it is between 20.0°C and 24.0°C in winter [1].

So far, most Heating, Ventilation and Air Conditioning (HVAC) systems for residential buildings usually employ a single-zone, On/Off control method which is rather simplistic [2]. Corresponding to the increasing demands for environment, energy, comfort and productivity, intelligent control methods are applied for improving thermal conditions in residential buildings [3,4]. Fuzzy control [5] is another type of intelligent control method. Comparing with classical ones, especially like Proportional Integral Derivative control (PID) that is widely used in industrial process control [6,7] due to its simplicity of structure, low-price, relative effectiveness and the familiarity of engineers, but cannot provide good enough performance in highly complex process controlling, fuzzy control can theoretically cope with complex processes [8] and is able to combine the advantages of PID control with human operator experience.

In this work, we first investigate the thermal dynamics of a building. Then a fuzzy control scheme with a meta-heuristic optimization algorithm called CLONALG, is proposed for the heating system of a residential building. This control system can make intelligent decisions of what magnitude of power the physical heating system should adopt at each time step based on the present indoor and outdoor temperatures. Due to empirical picking of fuzzy parameters initially, the target of CLONALG is to optimize these parameters to improve the performance of the fuzzy control system. The remainder of this paper is organized as follows. Section 2 describes the mathematical building thermal model. Section 3 presents the fuzzy controller used to control the heating system. Section 4 provides details about CLONALG algorithm. Section 5 explains the system design and formalizes the fuzzy system optimization process. Experimental results and analysis are given in Section 6. Finally, we conclude in Section 7.

2 Building Thermal Model

The room temperature is affected not only by auxiliary heating/cooling systems and electric appliances, but also by the solar radiation and the outside temperature. According to Achterbosch *et al.*[9], the heat balance of a building can be expressed as

$$\phi_h(t) + \phi_s(t) = \phi_t(t) + \phi_c(t) \quad (1)$$

where ϕ_h is the heat supplied by all internal heat sources; ϕ_s is the heat gained by solar radiation; ϕ_t is the heat loss through external contact; ϕ_c is the heat retained by the building.

The thermal system of the building can be expressed by Equations (2) - (6):

$$\frac{dT_w}{dt} = \frac{A_w}{C_w} \left[U_{wi}(T_{ai} - T_w) + U_{wo}(T_{ao} - T_w) \right] \quad (2)$$

$$\frac{dT_f}{dt} = \frac{A_f}{C_f} \left[\frac{pQ_s}{A_f} + U_f(T_{ai} - T_f) \right] \quad (3)$$

$$\frac{dT_c}{dt} = \frac{A_c}{C_c} \left[U_c(T_{ai} - T_c) \right] \quad (4)$$

$$\frac{dT_{ip}}{dt} = \frac{A_{ip}}{C_{ip}} \left[\frac{(1-p)Q_s}{A_{ip}} + U_{ip}(T_{ai} - T_{ip}) \right] \quad (5)$$

$$\begin{aligned} \frac{dT_{ai}}{dt} = \frac{1}{C_{ai}} & \left[Q_p + Q_e + (A_g U_g + U_v)(T_{ao} - T_{ai}) \right. \\ & + A_w U_{wi}(T_w - T_{ai}) + A_f U_f(T_f - T_{ai}) \\ & \left. + A_c U_c(T_c - T_{ai}) + A_{ip} U_{ip}(T_{ip} - T_{ai}) \right] \end{aligned} \quad (6)$$

In above equations: Q_e is the heat gained by using electrical equipments, Q_s is the solar radiation through glazing, Q_p is the heat supplied by the heating system, T_{ao} is the outside air temperature, T_{ai} is the inside air temperature, U is the thermal transmittance, C is the thermal capacitance, A is the area of the component, p is the fraction of solar radiation entering floor, and w, ip, f, c means external wall, internal partition, floor and ceiling respectively.

The area of each component is known after choosing the physical building model, and the properties of different building materials can be obtained from ASHRAE Handbook [10].

3 Fuzzy Logic Controller

Fuzzy Logic Controllers (FLC) have gained more and more prominence in recent years because of its ability to control devices which imitate the decision making of human being. Moreover, a FLC is efficient to cope with continuous states with the help of membership function (MF) and IF-THEN rules. In general, a FLC contains four parts: fuzzifier, rules, inference engine and defuzzifier. Firstly, a crisp set of input data is gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions. This step is known as fuzzification. Afterwards, an inference is made based on a set of rules. Lastly, the resulting fuzzy output is mapped to a crisp output using the MFs in the defuzzification step.

Specifically, in aforementioned building model the inputs include 4 elements: Q_p , Q_e , Q_s and T_{ao} and in order to simplify the problem, let's assume that Q_e and Q_s are both constant. T_{ao} can be simulated by using former weather data. Hence, the variable we need to control by our FLC is Q_p , which is the input of the building model but the output of the FLC. We define eT_{ai} as the error between the indoor temperature T_{ai} and the setpoint T_{set} , and eT_{ao} as the error between the outdoor temperature T_{ao} and T_{set} . Setpoint is the comfortable temperature that occupants prefer. To set the input variable(s) of the FLC there are two options: one is to consider eT_{ai} solely, like common air-condition, which is naive but still possible; the other one is to take eT_{ai} and eT_{ao} into account together, which gathers more information and therefore performs better. In our study, we prefer the latter. Therefore, we have two input variables, eT_{ai} and eT_{ao} separately and one output variable, Q_p .

In practice, there are different forms of MFs such as triangular, trapezoidal, piecewise linear, Gaussian, singleton, etc. They are curves which define how each crisp input point is mapped to a degree of membership between 0 and 1. Actually, these functions can be arbitrary curves whose shapes suit us from the point of view of simplicity, convenience, speed or efficiency under the only condition of their value between 0 and 1. In our study we capitalize on the Gaussian symmetrical function (GMF), Z-shape function (ZMF) and S-shape function (SMF) [11] because of their smoothness and concise notation that each of them can be defined by two parameters. Each fuzzy linguistic variable is expressed by three MFs, namely negative, zero and positive.

In fact, choosing MF types is not a tough job which is often out of empirical analysis. However, it is difficult to choose optimal fuzzy parameters for these MFs to design an optimal FLC. Usually people do this empirically too. In this study, we will use an AIS algorithm to find a near optimal set of parameters for the FLC. The proposed method involves arbitrarily picking an initial set of parameters and then finding a set of near optimal parameters by shifting the peak locations and tuning the deviations of fuzzy sets of antecedent MFs and consequent MF. We will discuss how to implement it in detail in Section 5.

4 Artificial Immune System Architecture

It has been proved that the human adaptive immune system possesses three capabilities: recognition, adaptation and memory [12]. When the human body is invaded by a specific pathogen or antigen, it will be recognized and bound by specific immunoglobulins or antibodies, which are secreted by B cells, to be tagged for attack by other part of the immune system or neutralised to death.

Figure 1 shows the antigen recognition and clonal selection process. An antibody, Ab, can recognize and bind an antigen, Ag, when Ab matches the structure of Ag. The regions of the antibodies that match the antigens are called paratopes, while the counterpart regions of the antigens are called epitopes. In this figure, Ab1 can match Ag1 but not Ag2, while Ab2 can neither match Ag1 nor Ag2, so Ab1 has higher affinity than Ab2 for encountering antigens. Higher affinity means higher probability of being selected and higher strength of clone and mutation. By continuous cloning and mutating existing ones, new generation of antibodies will be produced and among them new types of antibodies which may better match existing or new antigens are generated, for example Ab1' which can match both Ag1 and Ag2. This presents the adaptation capability of the immune system. Even if all antigens are destroyed, some relevant B cells will differentiate into memory cells. Therefore, if the same antigens reappear, the immune response will act sooner.

Inspired by the properties of human immune system, a variety of algorithms, such as Negative Selection, Clonal Selection, Immune Networks, and Dendritic Cell, have been designed to tackle different problems. The CLONALG algorithm [12], which belongs to Clonal Selection, we use to search the near optimal fuzzy parameters for the FLC is described below:

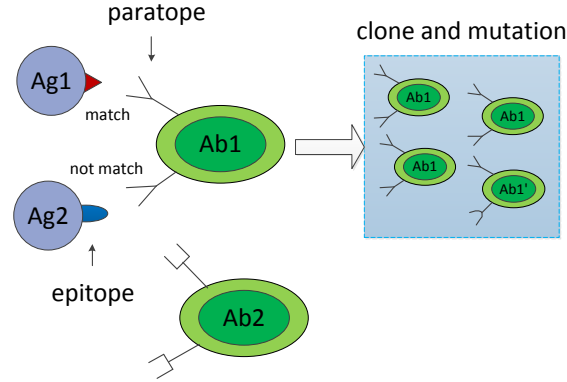


Fig. 1. Antigen Recognition, Clone and Mutation of Antibody

- (1) Generate a set (P) of candidate solutions, composed of the subset of memory cells (M) added to the remaining (Pr) population ($P = Pr + M$);
- (2) Determine (Select) the n best individuals of the population (P_n), based on an affinity measure;
- (3) Reproduce (Clone) these n best individuals of the population, giving rise to a temporary population of clones (C). The clone size is an increasing function of the affinity with the antigen;
- (4) Submit the population of clones to a hypermutation scheme, where the hypermutation is proportional to the affinity of the antibody with the antigen. A matured antibody population is generated (C^*);
- (5) Re-select the improved individuals from C^* to compose the memory set M. Some members of P can be replaced by other improved members of C^* ;
- (6) Replace d antibodies by novel ones (diversity introduction). The lower affinity cells have higher probabilities of being replaced.

5 System Design and Optimization

Based on the aforementioned model and technique, in this section we discuss the system design and the optimization of the fuzzy system. At every certain time interval, the thermal sensors of the building can record indoor and outdoor temperatures and sent them as inputs to the fuzzy controller. According to the MFs and rules of the fuzzy controller, after the fuzzifier-inference-defuzzifier process the physical heating appliance in the building will be notified a magnitude of heating power. Because the MFs defined empirically can not perform very well, therefore optimizing the fuzzy controller is a must step and this is the target of the artificial immune system, which in this application is a meta-heuristic

algorithm named CLONALG. For the reason that the variation of outdoor temperature is continuous and rather slow, we can capitalize on a specific sinusoidal curve to simulate one day's outdoor temperature variation, and use CLONALG to tune fuzzy controller to make good decisions for general real-time weather situations.

Now we move on to this optimization problem formalization. Assume that there are m input variables $[x_1, x_2, \dots, x_m]$ and one output variable y . The total number of fuzzy sets N is calculated as follows: $N = \sum_{i=1}^m n_i + n_o$, where m is the number of input variables, n_i and n_o are the number of fuzzy sets for i th linguistic input variable and the linguistic output variable. A set P with size of $2N$ contains the peak location and deviation of every fuzzy set, that is: $P = [\mu_{in}, \sigma_{in}, \mu_{out}, \sigma_{out}]$, where $\mu_{in} = [\mu_1^1, \mu_2^1, \dots, \mu_{n_1}^1, \dots, \mu_{n_i}^i]$, $\sigma_{in} = [\sigma_1^1, \sigma_2^1, \dots, \sigma_{n_1}^1, \dots, \sigma_{n_i}^i]$, $\mu_{out} = [\mu_1^o, \dots, \mu_{n_o}^o]$, and $\sigma_{out} = [\sigma_1^o, \dots, \sigma_{n_o}^o]$, for all $i = 1, 2, \dots, m$. The objective of the method is to minimize the difference between the inference output y and the desired output y^* , in our case are controlled Q_p and desired Q_p separately, with respect to P : $C = \min_P (y - y^*)^2$, where: $y = f(x_1, x_2, \dots, x_m, P)$, and $y^* = f(x_1, x_2, \dots, x_m)$. We can see that the objective function C depends not only on P but also the inputs. In order to eliminate the dependence of the inputs, we use the Root Mean Square Error (RMSE), such that: $RMSE(y) = \sqrt{E((y_t - y_t^*)^2)} = \sqrt{\frac{\sum_{t=1}^T (y_t - y_t^*)^2}{T}}$, where T is the number of points of the whole trajectory. Therefore the objective function becomes: $C = \min_P \left[\alpha \sqrt{\frac{\sum_{t=1}^T (y_t - y_t^*)^2}{T}} \right]$. All else being known, at a time t indoor temperature only depends on the output power of the heating system (we can see this in Equation (6)). Therefore at every time t , indoor temperature is a function of the output power of the heating system, recorded as: $T_{ai}^t = g^t(Q_p^t)$. Moreover, because $g^t(\cdot)$ is linearly monotonically increasing, the final objective function can be expressed as follows: $C = \min_P \left[\alpha \sqrt{\frac{\sum_{t=1}^T (g^t(y_t) - g^t(y_t^*))^2}{T}} \right]$. Hence, after the minimization process, the FLC with fuzzy parameters in P is optimized.

6 Experiments

We first empirically pick μ and σ for all MFs of input and output variables. Then these parameters are optimized by CLONALG. Due to CLONALG can not guarantee to obtain optimal values, we run CLONALG for 30 times and take their mean values as near-optimal parameters for the fuzzy controller: $\mu_{in} = [0, -0.4, 0, 0, -0.54, 0]$, $\sigma_{in} = [-0.61, 0.1, 0.508, -6.952, 5.333, 6.426]$, $\mu_{out} = [22, 18, 22]$, and $\sigma_{out} = [10.89, 1.889, 26.24]$. In Figure 2(a), actual recorded weather data obtained from EERE [13] is used as the outdoor air temperature, which is depicted by a dashed blue line. From the simulation result, it can be found that during this period, the indoor temperature, which is delineated by a green line, is able to be kept at $22^\circ C$. Even during the first three days' extremely cold weather, the indoor temperature is retained at the setpoint. Figure 2(b) shows the amplification of the room temperature, and one can see that the variation of

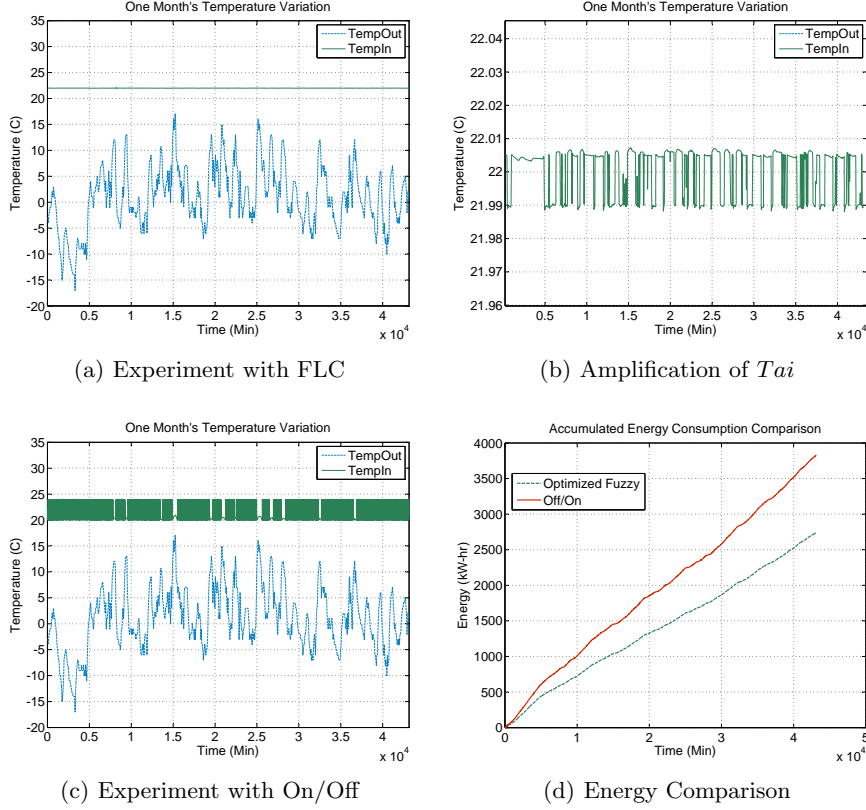


Fig. 2. Experimental Results

this temperature is almost within $\pm 0.01^{\circ}C$. Moreover, the simulation result with On/Off control is described in Figure 2(c). For this control method, the heating system turns on when the room temperature is below $20^{\circ}C$, while it turns off when above $24^{\circ}C$. In order to keep a comfort temperature, the heating system has to turn on and off frequently, which will jeopardize the physical system and reduce its service life. Finally, the accumulated energy consumption comparison between the optimized FLC and the On/Off control is shown in Figure 2(d). We can see that compared with the On/Off control which uses $3830 \text{ kW} \cdot \text{hr}$ in total, the optimized FLC uses $2742 \text{ kW} \cdot \text{hr}$ in total, so that it consumes $1088 \text{ kW} \cdot \text{hr}$ less energy.

7 Conclusion

This paper has presented a fuzzy logic controller optimized by an artificial immune system algorithm to keep thermal comfort while consuming less energy in

residential buildings. The experimental results show that by employing this controller, the indoor temperature can be more stable and thus more comfortable than the classical On/Off control and consumes less energy. However, the work conducted here is still a preliminary step towards a completely autonomous HVAC system. In future work, the comparison with other optimization algorithms like PSO will be made. Furthermore, other systems such as a lighting system and a ventilation system, will be taken into account together. Certainly this is also a good application for multi-agent paradigm. Based on the multi-agent framework, agent-to-agent communication, cooperation and coordination can be employed to provide a more comfortable residential environment and consume less energy.

References

1. ISO7730: Ergonomics of the thermal environment - analytical determination and interpretation of thermal comfort using calculation of the pmv and ppd indices and local thermal comfort criteria (2005)
2. Paris, B., Eynard, J., Grieu, S., Polit, M.: Hybrid pid-fuzzy control scheme for managing energy resources in buildings. *Applied Soft Computing* **11**(8) (2011) 5068 – 5080
3. Moon, J.W., Kim, J.J.: Ann-based thermal control models for residential buildings. *Building and Environment* **45**(7) (2010) 1612 – 1625
4. Zhu, J., Lauri, F., Koukam, A., Hilaire, V., Simoes, M.G.: Improving thermal comfort in residential buildings using artificial immune system. In: 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing(UIC). (Dec 2013) 194–200
5. Kruse, R., Gebhardt, J.E., Klowon, F.: Foundations of Fuzzy Systems. John Wiley & Sons, Inc., New York, USA (1994)
6. Kaya, İ., Tan, N., Atherton, D.P.: Improved cascade control structure for enhanced performance. *Journal of Process Control* **17**(1) (January 2007) 3–16
7. Thomas, B., Soleimani-Mohseni, M., Fahl, P.: Feed-forward in temperature control of buildings. *Energy and Buildings* **37**(7) (2005) 755 – 761
8. Menon, R., Menon, S., Srinivasan, D., Jain, L.: Fuzzy logic decision-making in multi-agent systems for smart grids. In: Computational Intelligence Applications In Smart Grid (CIASG), 2013 IEEE Symposium on. (2013) 44–50
9. Achterbosch, G., de Jong, P., Krist-Spit, C., van der Meulen, S., Verberne, J.: The development of a convenient thermal dynamic building model. *Energy and Buildings* **8**(3) (1985) 183 – 196
10. ASHRAE: ASHRAE Handbook: Fundamentals. American Society of Heating, Refrigerating, and Air-Conditioning Engineers (2005)
11. Mandal, S.N., Choudhury, J.P., Chaudhuri, S.: In search of suitable fuzzy membership function in prediction of time series data. *International Journal of Computer Science Issues (IJCSI)* **9**(3) (2012)
12. de Castro, L.N., Von Zuben, F.J.: Artificial immune systems: Part i–basic theory and applications. Universidade Estadual de Campinas, Dezembro de, Tech. Rep (1999)
13. EERE: Weather Data Golden-NREL 724666 (TMY3). Energy Efficiency & Renewable Energy, U.S. Department of Energy (2012)

Floods Trajectories Modeling and Dynamic Relief Planning: A Bees Foraging Approach

Kawther Hmaidi and Jalel Akaichi

University of Tunis, Institut supérieur de gestion, Bardo, Tunis
hmaidi.kawther@live.com, j.akaichi@gmail.com

Abstract. Natural disasters represent hazards resulting from extreme geophysical events. Floods particularly are one of the most occurring disasters. They affect annually different regions of the world with varying intensities causing materiel damages and fatalities. Despite the efforts done by rescue agents in this context, inefficiencies occur yet. Thus, the need of disaster management information systems is becoming critical to mitigate the effect of natural hazards. In this paper, we aim to provide a dynamic decision making tool inspired by the foraging behavior of honey bees which assists in managing relief operations and assigning rescue agents to affected areas. We propose, equally, a trajectory data warehouse model for flood tracking and affected areas location.

Keywords: Bees foraging; Dynamic allocation; Relief planning; Floods Trajectory data warehouse

1 Introduction

The period following natural disasters represents a critical period due to various challenging factors such as the large number of casualties at stake, the time required for evacuation and the lack of rescue officers and material resources. Hence, the need of disaster management information systems is becoming critical to deliver the right information to public authorities concerned by decision making. Despite the incredible efforts done in this respect, inefficiencies in relief activities occur yet. In fact, the majority of current Disaster management systems usually are mere information systems used for graphical representation of disaster relevant data. Nevertheless, there is no efficient means which allow quick analysis of disaster information and provide rapidly an adequate planning for relief agents allocation. Moreover, when the management of a disaster extends on large scale, decision making becomes more difficult and time becomes a critical factor.

The goal of this paper is to:

- Design a trajectory data warehouse model to track the changing position of floods water waves in order to locate affected areas.
- Organize dynamically the work of rescue teams through a dynamic decision making tool inspired by the foraging behavior of honey bees.

The remainder of this paper is organized as follows: In section 2, we present some research works related to bees inspired algorithms and their applications. In section 3, we propose a conceptual model for floods trajectory data warehouse. In section 4, we describe the foraging behavior of bees in nature and we provide an algorithm inspired by the latter behavior for dynamic relief management. In section 5, we will summarize the work and propose new perspectives to be done in the future.

2 Related Works

2.1 Main algorithms inspired by bees foraging behavior and its applications

Over years, the computational researches have been increasingly interested to biological phenomena, as source of modeling paradigms. Particularly, the bees inspired algorithms has emerged recently and has proved its efficiency in several application domains. The main purpose was to mimic the efficiency of honey bees organization within its colonies either for establishing analogies for system functioning or for resolving renowned optimization problems. Several woks in the literature have dealt with bees inspired algorithms using different nomenclatures. We can cite briefly: bees system [5], [6], [9], Bee Colony Optimization [7], Bee Hive [11], Artificial Bee Colony [4] and Virtual Bee Algorithm [12].

The main domains are: Genetic Algorithm Improvement, Traveling Salesman Problem (TSP), Stochastic Vehicle Routing Problem, Dynamic Allocation of Internet Service, Job Shop Scheduling, Telecommunication Network Routing, Neural Network and Routing Protocol for Wireless Sensor Networks.

2.2 Models Inspired by Honeybees Foraging Behavior for Collective Decision Making in Relief Operations

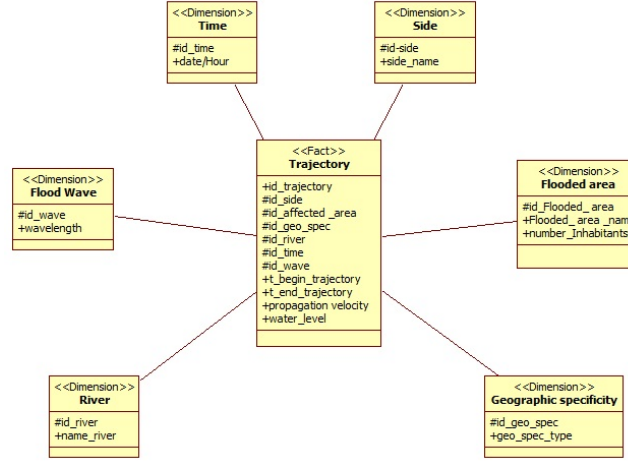
Inspired by the concept of collective decision making of honeybees, Chen and Pea-Mora proposed, in [3], a decentralized and collective decision making approach for large scale disasters. In fact, it enables immediate deployment of heavy construction equipment, which supports critical lifesaving activities during urban search-and-rescue period. As well, Aldunate and colleagues presented, in [1], another distributed collaborative decision-making model which allows the system communication without any commander. Indeed, a decision making problem is modeled as the selection of one best option among available options to perform a task.

3 Floods Trajectory Data Warehouse Model

The use of remote sensing technologies, sensors and localization systems opened the way to the applications exploiting the location. Hence, the huge volume of

generated trajectory data must be stored in a multidimensional model, called trajectory data warehouse. The latter, allows analysis and gives the possibility of extracting knowledge from historical data which ensures better decision making. Trajectories are stored as set points (X, Y, T) where the couple (X, Y) represents the space dimension and (T) represents the time dimension. In our context, a flood travels along a river as a wave with velocity and depth continuously changing and usually affects areas located close to it with variable levels [8]. Therefore, we consider water waves as the moving object which changes its position over space and time. Our model is presented by a star schema as illustrated below in Fig.1, composed by the fact table:Trajectory and dimensions tables:Time, Flooded area, Side, Geographic specificity, River and Flood wave.Indeed, the fact contains measures (water-level and propagation-velocity) representing analysis values while dimensions are defined as analysis axes.

Fig. 1. Floods trajectory data warehouse Star schema

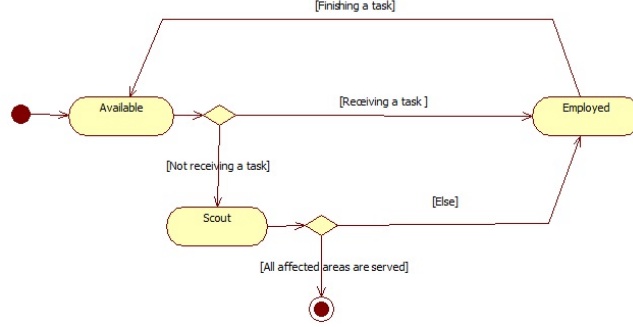


4 Dynamic Relief Planning

The aim of this work is to find an optimal allocation of relief agents to flooded areas in a dynamic way. We suppose that rescue agents are divided into homogeneous teams, where each team is composed of a predetermined number of agents with different rescue skills. Every relief team is equipped with a mobile device. Once a relief team finished an evacuation task, he changes its state to available and then either he indicates the number of evacuees or he indicates that this area does not request relief yet if all inhabitants are evacuated. Iteratively, considering areas not served yet and areas served but still requesting relief, the system has to choose which area to serve based on a probability with which relief teams

are assigned. In order to save the biggest number of evacuees, the relief demand may correlate highly with the number of survivals trapped in the flooded areas. During relief operations, a relief team may have the different states as illustrated below in Fig.2.

Fig. 2. Different states of a relief team



4.1 Mathematic Formulation of the Problem

The main goal of our model is to maximize the number of evacuees by assigning relief teams to flooded areas. Inspired by generalized assignment problem where the objective function aims to assign a set of tasks to a set of agents, we formulate the problem as an integer programming model with the objective to maximize the number of evacuees. The objective function is as follow:

$$\text{maximize} \quad \sum_{i=1}^n \sum_{j=1}^m E_{ij} x_{ij} \quad (1)$$

subject to

$$x_{ij} \in \{0, 1\} \quad (2)$$

$$\sum_{i=1}^n a_{ij} x_{ij} \leq b_j \quad \forall j : 1 \dots m \quad (3)$$

Where E_{ij} : Number of evacuated inhabitants in flooded area i by relief team j ;
 x_{ij} : Variable decision =1 if flooded areas i is assigned to relief team j , 0 otherwise;
 n : Number of flooded areas $i:1..n$; m : Number of relief teams $j:1..m$;
 b_j : Capacity of relief team j ; a_i : Relief resource needed by flooded area i .
 Constraint (3) specifies the maximum capability associated with each given relief team.

4.2 Bees Foraging Behavior in Nature

The foraging behavior of honey bees in nature designates the search of food. During this process, the forager bees are divided into employed: they are engaged in exploiting a food source, and unemployed: They are continually at look out for a food source to exploit. There are two types of unemployed foragers: scouts: searching the environment surrounding the nest for new food sources and onlookers: waiting in the nest and establishing a food source through the information shared by employed foragers. The communication is insured through the waggle dance in order to recruit the nest mates and send more follower bees to more promising patches [2].

4.3 Correspondence between Honeybees Foraging Behavior and Relief Agents Allocation

At the outset of our research, it was immediately obvious that the relief teams allocation and foragers allocation problems were similar. We note that the set of relief teams having to be assigned to flooded areas is analogous to bees foragers allocated to multiple flower patches. Both of them aim to choose the most profitable sources in term of either flowers nectar or number of trapped people in affected areas. Table 1 illustrates the previous correspondence.

Table 1. Analogies between honey bees foraging behavior and relief agents assignment

Bees foraging behavior	Relief teams allocation
<ul style="list-style-type: none"> • Food sources • Fitness of food sources: Nectar amount 	<ul style="list-style-type: none"> • Flooded areas • Severity degree of affected areas : Number of trapped inhabitants
<ul style="list-style-type: none"> • Forager bees • Employed bees • Onlookers bees • Scout bees 	<ul style="list-style-type: none"> • Relief teams • Relief teams assigned to flooded areas • Available relief teams • Relief teams finishing a task and not receiving a new one
<ul style="list-style-type: none"> • Objective: Maximize the total nectar amount collected from different flower patches 	<ul style="list-style-type: none"> • Objective: Maximize the total number of evacuees from different affected areas by assigning relief team i to flooded area j

4.4 Dynamic Relief Allocation Algorithm

As detailed previously, the aim of the proposed model is to dynamically allocate rescuers. Hence, the algorithm Relief-Alloc has to check, iteratively, the list and choose the flooded areas to serve in order to maximize the total number of evacuees. For each area, a probability with which relief teams are assigned is computed by the procedure Compute-probability. Thus, as the number of trapped

persons increases, the number of assigned relief teams increases too.

Algorithm Relief-Alloc

Input: Nbr flooded areas , Nbr relief teams, Nbr trapped people in area i
Output: Dynamic allocation of relief teams to flooded areas
Begin
While (List of affected area not empty)
For each flooded area i
Compute probability()
End for
For each relief team j
If state j = available then
Assign available relief teams to flooded areas having max pi
Set state of assigned relief teams to employed
End if
End for
If task accomplished then
Share information about number of evacuees
Set state of relief team to available
End if
Compute the number of remainder trapped people in each served area i
If number of remainder trapped people = 0 then
Remove areas from list of flooded areas
Else
Update list of flooded areas
End if
End while
End

Procedure Compute-probability

Input:L: list flooded areas, n: Nbr flooded area, nbrtr : Nbr trapped people
Output: Lprob: list flooded areas with relative p
Begin
For each affected area

$$\frac{nbrtr_i}{\sum_{i=1}^n nbrtr_i} \quad (4)$$

End for
End

5 Conclusion and Future Work

Floods have been considered as one of the most devastating disasters throughout the last century either in terms of property damage or human casualties. Thus, in order to improve relief operations and save the biggest number of victims, our work focused on analyzing floods data and organizing the work of relief teams. By drawing analogies from the foraging behavior of honey bees in nature, we have provided an algorithm to dynamically allocate relief to flooded areas. We have equally modeled the trajectories of floods through a trajectory data warehouse schema. As part of our future work, we aim to include fuzzy logic to locate flood waves positions. As well, we propose to improve the allocation of relief by considering the nearest flooded zones to choose when assigning rescuers

References

1. Aldunate, R.G., Pea-Mora, F., Robinson, G.E.: Collaborative Distributed Decision Making for Large Scale Disaster Relief Operations: Drawing Analogies from Robust Natural Systems. *J.Complexity*.11, 28-38(2005)
2. Bitam, S., Batouche, M., Talbi, E.L.G.: A Survey on Bee Colony Algorithms. In: 2010 IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW), pp.1-8, Atlanta (2010)
3. Chen, A.Y., Pea-Mora, F.: Decentralized Approach Considering Spatial Attributes for Equipment Utilization in Civil Engineering Disaster Response. *J.Comput.Civ.Eng.*25,457- 470(2005)
4. Karaboga, D., Basturk, B.: Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In:12th International Fuzzy Systems Association World Congress, IFSA 2007, pp.789-798 , Cancun (2007)
5. Lucic, P., Teodorovic, D.: Bee system: Modeling Combinatorial Optimization Transportation Engineering Problems by Swarm Intelligence. In:Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, pp.441-445, Azores Islands(2001)
6. Lucic, P., Teodorovic D.: Transportation Modeling: An Artificial Life Approach. In:ICTAI'02 14th IEEE International Conference on Tools with Artificial Intelligence, pp.216-223(2002)
7. Markovi, G.Z., Teodorovi, D.B., Aimovi-Raspopovi, V.S.: Routing and wavelength assignment in all optical networks based on the bee colony optimization. *J.AI Communications*.20, 273-285(2007)
8. Mujumdar, P.P.: Flood wave propagation. *J.Resonance*.6, 66-73 (2001)
9. Sato, T., Hagiwara, M.: Bee System: Finding Solution by a Concentrated Search. In: IEEE International Conference on Systems, Man, and Cybernetics, pp.3954- 3959, Orlando (1997)
10. Teodorovic, D., Davidovic, T., Selmic, M.: Bee Colony Optimization: The Applications Survey. *J.ACM Transactions on Computational Logic*.12, 1-20 (2011)
11. Wedde, H.F., Farooq, M., Zhang, Y.: BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior, Ant Colony and Optimization and Swarm Intelligence. In: 4th International Workshop, ANTS 2004, pp. 83-94, Brussels (2004)

12. Yang, X.S.: Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms. In: First International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2005, pp.317-323, Spain (2005)

Improving Biogeography Based Optimization by Using Stochastic Local Search and Solving MI-FAP Problem in GSM Networks

Yasmine Lahsinat and Dalila Boughaci

LRIA-FEI- USTHB
Computer Science Department
BP 32 El-Alia Beb-Ezzouar, Alger 16111
ylahsinat@usthb.dz; dboughaci@usthb.dz

Abstract. In this paper, three hybrid methods based on both Biogeography Based Optimization (BBO) and Stochastic Local Search (SLS) are studied for the Minimum Interference Frequency Assignment Problem in GSM networks (MIFAP). Experiments on various benchmark problems are performed to show and compare the effectiveness of our approaches. The comparisons between the hybrid BBO and the standard BBO and Genetic Algorithms(GA) show the benefits of the proposed improvements in BBO for the MI-FAP.

Keywords: Frequency Assignment Problem, Biogeography Based Optimization, Hybrid, Stochastic Local Search, MI-FAP, GSM networks.

1 Introduction

The Minimum Interference Frequency Assignment Problem in GSM networks (MI-FAP) is the problem of finding an assignment of a small number of frequencies to a large number of transceivers that minimizes the interferences level. The MI-FAP is known to be NP-Hard[2]. The MI-FAP can be stated as follows: Let us consider a GSM network N composed of a set of cells: $C = \{C_1, C_2 \dots C_i, \dots C_{nc}\}$, where nc is the number of cells. Each cell required a number of frequencies which is defined by the number of transceivers TRX s or traffic demand $D_i = \{1, 2 \dots, x\}$. Each network has its own interval $[F_{min} - F_{max}]$ of operational frequencies with: $|F_{min} - F_{max}| = NF$ being the number of frequencies. For each cell, we have a set of operational frequencies, which means that some cells can not use some frequencies. The co-channel and adjacent channel constraints are generally represented by matrix: Let consider the $MCo[nc][nc]$ matrix :Where one element $MCo[i][j]$ corresponds to the rate of co-channel interference in case one TRX of cell j uses the same channel (f) of the cell i , and the $MAdj[nc][nc]$ matrix: where one element $MAdj[i][j]$ corresponds to the rate of the adjacent channel interference in case : one TRX of cell j uses a adjacent channel to one TRX of cell i (f), ($f + 1/ f - 1$).

Given a list of TRX , an interval of operational frequencies, a matrix of Co-channel interference, a matrix of adjacent channel interference, the goal is then to

assign a frequency to each *TRX*s with minimum co-channel and adjacent channel interferences. The solution of the problem is denoted by $Sol = \{f_1, f_2, f_i \dots f_d\}$, where d is the number of the *TRX*s of the network. Every element containing in Sol correspond to a frequency f , assigned to a j^{th} *TRX* included in cell i . The MI-FAP is the problem of finding an assignment for each element of Sol that minimizes the cost function given in formula (1) and proposed in [3].

$$\text{Minimize} \quad \sum_{i=1}^{NC} \sum_{j=1}^{NC} (Y_i \times Interfco(i, j) + Z_i \times Interfadj(i, j)) * w_i \quad (1)$$

The objective function (1) minimizes the overall interferences which equal the total co-channel interferences and the total adjacent channel interferences, taking into account the traffic over a cell: Where w_i is a weight associated to Cell i due to amount of traffic carried. Y_i takes the value 1 if *TRX* from cell i and *TRX* from cell j operates on the same frequency, 0 otherwise. $Interfcoij$ is an element in MCo : if *TRX* from i and *TRX* from j , operates on the same frequency, $Interfcoij$ is equal to the rate of the interferences caused by this assignment. Z_i takes the value 1 if *TRX* from cell i and *TRX* from cell j operates on adjacent frequencies, 0 otherwise. $Interfadj$ is an element in $MAAdj$, if *TRX* from cell i and *TRX* from cell j operates on adjacent channel, so $Interfadj$ is equal to the rate of the interference generated by this assignment. Various methods and approaches are proposed to address the MI-FAP problem, including the following: Improved tabu search algorithm [8], Genetic tabu search [7]. Evolutionary approaches such as EAs and Ant Colony optimization algorithms (ACO/EAs)[5].hybrid approaches[6]. In this paper, we propose three combinations of BBO with SLS for the MI-FAP in GSM networks. The rest of this paper is organized as follows: Section 2 details the proposed methods for solving MI-FAP. Section 3 presents some results of various tests and experimentations. Finally, Section 4 concludes and gives some future works.

2 Biogeography Based Optimization for MI-FAP

Biogeography Based Optimization is an evolutionary algorithm for global optimization [9] proposed by *Dan Simon* in 2008. The philosophy of this algorithm is derived from the mathematical models of biogeography. The mathematical models were introduced in the 1960s by Robert MacArthur and Edward Wilson. These models describe the migration process (emigration and immigration) of species from one island to another[9]. BBO is a population based algorithm where each individual represents a habitat. Habitats quality is assessed using the Habitat Suitability Index (HSI) which determines the fitness of a solution. Each habitat is characterized by a set of variables called Suitability Index Variable (SIV) and a number of species which are related to the HSI value. Each habitat is defined by two rates, i.e the emigrating and the immigrations rates, which are function of the number of species and can be calculated as follows:

$$\lambda = I(1 - k/n) \quad (2)$$

$$\mu = E(k/n) \quad (3)$$

Where E , I are respectively the maximum emigration and immigration rates, k is the actual number of species in the habitat, and n is the maximum number of species which a habitat could have.

Figure 1 shows the relationship between the fitness islands, emigration rate μ and immigration rate λ . The $S1$ represents a poor solution and $S2$ a good solution. In BBO, good solution like $S2$ is characterized by a high HSI (Habitat suitability index), a large number of species, a high emigration rate μ and a low immigration rate λ . On the other hand, poor solution like $S1$ has a high immigration rate and low emigration rate. The population evolves through

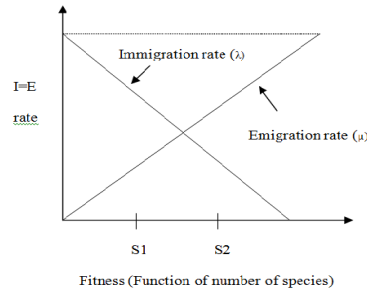


Fig. 1. The relationship between the fitness habitats (number of species) emigration rate μ and immigration rate λ

the two main processes, i.e. migration and mutation. In this section, we propose a BBO algorithm to solve the MI-FAP problem. The main background of our approach is given in the following.

2.1 Solution Representation

The potential solution which is the frequency plan is represented by a vector Sol with d elements where d is the total number of $TRXs$. Each frequency is represented by a number from 1 to NF . Each element from Sol represents the frequency assigned to the K^{th} TRX of the cell i . $Sol \leftarrow \{F_1, F_2 \dots F_d\}$.

2.2 The Migration process

The migration in BBO is probabilistic process that allows data sharing between solutions. Indeed, the good solutions share their features SIV (variables values) with the bad ones in order to improve their quality. For that purpose, different variants of migration process have been proposed[4] but the main idea is to use the emigration and immigration rates to identify potential migrants (one

immigrant and one emigrant). As part of our work, we have chosen the Partial-Based Immigration BBO. This process uses the emigrating rate to select the good solution and the immigrating rate to identify the bad one.

2.3 The Mutation process

In BBO, like other evolutionary algorithms, the mutation is used to increase diversity among the population. Within our work, we used a less complex procedure than that of Dan Simon[9].

2.4 The standard BBO for the MI-FAP

The process can be summerized as follows: in BBO, a series of habitats, in our case frequency plan are randomly generated; then for one generation we used the following two step method : for each habitat we assessed the fitness value (the HSI value) and sorted the population from the best solution to the worst one. In the next step, we mapped the HSI to the number of species to calculate the immigration rate λ and the emigration rate μ (following the above mentioned equations (2) and (3), respectively). Then we apply the migration. Mutate the lower half of the population using a mutation parameter which will be assigned a fixed value. Evaluate the population for the next generation of BBO. The process is repeated until the stopping criterion is met.

The pseudo code is sketched in Algorithm 1.

Algorithm 1 : The BBO for the MI-FAP.

Require: an MI-FAP instance, HSI
Ensure: The best solution found x
1: Generate a random Population
2: **while** (Stopping criterion is not met) **do**
3: Find the HSI for each habitat
4: Compute the number of species according to the HSI value
5: Compute μ and λ values
6: Apply the Migration process
7: Apply the mutation process
8: **end while**

2.5 The Hybrid BBO for the MI-FAP

In this section, we propose three hybrid methods denoted BBOSLS (1), BBOSLS(2) and BBOSLS(3) that combine the biogeography based optimization(BBO) and the Stochastic Local Search(SLS). The proposed methods start with a randomly generated population, followed by an evaluation and migration steps as done in classical BBO. Then the SLS routine is performed with three different ways. We propose to apply the SLS instead of the mutation on different member from the BBO population. The SLS step in the three hybrid BBO/SLS methods is an efficient local search. SLS explores efficiently the neighbor search of the solution in order to locate high quality solution. The process mixing between BBO and

SLS is repeated until stopping criterion is met. The main contributions of this work, on the BBO hybridization, are summarized in the following but first we present the Stochastic Local Search (SLS) used with the three methods.

The stochastic Local Search The simple Stochastic Local Search used here is inspired from the one used in[1]. The SLS technique starts with a randomly generated solution x and tries to find better solution in the current neighborhood. The neighboring solution x' of the solution x is obtained by modifying the frequency assigned to a selected TRX . The perturbed TRX is selected according to one of the two following criteria[1] : The first criterion consists in choosing the TRX in a random way with a fixed probability $wp > 0$. The second criterion consists in choosing the conflicting node: the routine proposed here looks after the assignment that caused the most interference with its neighbors and we move it, by assigning to it the less used frequency. The process is repeated until we met stopping criterion.

The BBOSLS(1) algorithm for MI-FAP the SLS procedure-replacing the mutation- is the next step after the migration process. We define a parameter (with the same value as $Pmutate$) to decide whether to apply SLS or not on a solution from the BBO population. The pseudo code is sketched in Algorithm 2.

Algorithm 2 : The BBOSLS(1) for the MI-FAP.

Require: an MI-FAP instance, HSI
Ensure: The best solution found x

```

1: Generate a random Population
2: while (Stopping criterion is not met) do
3:   Find the HSI for each habitat
4:   Compute the number of species according to the HSI value
5:   Compute  $\mu$  and  $\lambda$  values
6:   Apply the Migration process
7:   for (Each solution) do
8:      $r \leftarrow$  random number between 0 and 1;
9:     if ( $r < Pmutate$ ) then
10:      Apply the SLS process
11:     else
12:      Not apply
13:     end if
14:   end for
15: end while

```

The BBOSLS(2) algorithm for MI-FAP the hybridization of BBO consists on replacing the mutation by SLS and applying it, right after the migration, on the half worst part of the current population. The process tries to improve the worst solutions in the population. The pseudo code is sketched in Algorithm 3.

Algorithm 3 : The BBOSLS(2) for the MI-FAP.

Require: an MI-FAP instance, HSI

Ensure: The best solution found x

```

1: Generate a random Population
2: while (Stopping criterion is not met) do
3:   Find the HSI for each habitat
4:   compute the number of species according to the HSI value
5:   compute  $\mu$  and  $\lambda$  values
6:   Apply the Migration process
7:   for (Each solution from  $PopSize/2$  which represents the half worst of the population) do
8:     Apply the SLS method
9:   end for
10: end while

```

The BBOSLS(3) algorithm for MI-FAP we apply the SLS process on the best solution contained in the population, after the migration process. The pseudo code is sketched in Algorithm 4.

Algorithm 4 : The BBOSLS(3) for the MI-FAP.

Require: an MI-FAP instance, HSI

Ensure: The best solution found x

```

1: Generate a random Population
2: while (Stopping criterion is not met) do
3:   Find the HSI for each habitat
4:   Compute the number of species according to the HSI value
5:   Compute  $\mu$  and  $\lambda$  values
6:   Apply the Migration process
7:   Apply the SLS method on the Best solution of the current population
8: end while

```

3 Experiments and numerical results

We carried several experiments to evaluate the performance of BBO and its variants on the MI-FAP and compared it with GA. The programs are written in java language and run on an Intel core i5 with 4GB of RAM. We use the realistic data COST259 given by [10]. Due to the non-deterministic nature of the algorithms, 10 runs have been considered for each algorithms. To make a fair comparison between the proposed approaches, the running time is fixed to: 1800 second. The adjustment of the different parameters of the proposed approaches is fixed by an experimental study. The BBO parameters are: the Population size which is set to 150, the Elitism =2 and the mutation rate=0.1. The SLS parameters are: $w_p=0.25$, $timeout=300$ seconds for the three methods BBOSLS(1), BBOSLS (2) and BBOSLS(3). The parameters of GA are: the Population Size is set to 150, the crossover rate $P_{cross}=0.9$, the mutation rate =0.05. The GA method is a genetic algorithm that uses crossover and standard mutation operators without a local search.

Table 1 gives some results of SBBO, BBOSLS(1), BBOSLS(2), BBOSLS(3) and GA. The experimental study shows that BBO is able to find solutions that are very close to the one found by the GA. Comparing the hybrid developed

methods, we see that the BBOSLS(2) performs poorly on Siemens instances and K instance compared to the BBOSLS(1) and BBOSLS(2). The solution obtained by BBOSLS(3) is better than those obtained with both BBOSLS(1) and BBOSLS(2) hybrid methods. The comparison between the standard BBO and BBOSLS(3) shows clearly that the hybrid approach perform better for all the *Siemens* and K instances. We can conclude that the SLS process enhances the quality solution of the BBO algorithm.

Table 1. SBBO vs. GA vs. Hybrid methods BBOSLS where timeout = 1800 second

Instances	Statistical Features	SBBO	BBOSLS(1)	BBOSLS(2)	BBOSLS(3)	GA
Siemens 01	Best	223,62	32.19	100.96	3.17	140.682
	Mean	250,18	44.93	104.40	10.43	157.230
	Worst	270,13	105.06	110.72	30.52	183.738
	Std.	13.22	22.06	3.174	8.204	14.698
Siemens 02	Best	5941,29	5204.90	6325.19	3543.00	3778.611
	Mean	6426,40	5527.35	6471.71	4501.81	5009.435
	Worst	7068,98	5904.40	6533.86	5303.90	6915.863
	Std.	372.10	209.22	61.53	557.45	963.669
Siemens 03	Best	1460,65	1102.76	1445.11	825.31	1136.922
	Mean	1544,94	1202.81	1498.99	907.54	1276.527
	Worst	1771,72	1282.29	1617.58	1054.94	1575.488
	Std.	94.79	63.72	47.30	79.06	134.455
Siemens 04	Best	108401,08	97204.81	110998.65	66463.95	101010.6
	Mean	110830,3	101117.4	112104.6	74343.99	104288.8
	Worst	114803,53	103967.9	112895	90094.98	111591.1
	Std.	1957.60	1978.70	714.79	6986.715	2976.158
Tiny	Best	0	0	0	0	0
	Mean	0	0	0	0	0
	Worst	0	0	0	0	0
	Std.	0	0	0	0	0
Swisscom	Best	0	0	0	0	0
	Mean	0	0	0	0	0.659
	Worst	0	0	0	0	1.553
	Std.	0	0	0	0	0.525
K	Best	98.26	24.30	108.94	15.97	32.900
	Mean	106.27	33.99	118.29	28.303	40.788
	Worst	125.53	42.31	131.27	86.46	47.516
	Std.	8.21	6.16	7.53	20.97	4.318

We present in Table 2, a comparison with some well-known methods for the different instances found in [10]. The value cost reported in Table 2 for the different approaches is the best minimum sum over all Co- and adjacent channel interferences occurring between pairs of cells recorded from 10 runs. The column other methods presents the result obtained with other techniques found in [10]. We compared our method with Simulated Annealing combined with Dynamic Programming (SA-DP), Original Assignment (OA), Dynamic Tabu Search and Threshold Accepting (DTS-TA)[10]. As mentioned in [10] in most cases, for the result reported, the run-time is, in fact, not known: typically the methods have a run-time of a few to several hours on a modern PC. Indeed the proposed model took more constraints into account and the objective was to find a feasible solution. As shown in Table 2, our methods succeed in finding the optimum for the Tiny and Swisscom. For the Siemens 1, the BBOSLS(2) is better than the well-known SA-DP method. However, for the other benchmarks our methods fail to find good results compared to the other methods. We note that our methods are launched 30 minutes maximum. We believe that our methods will give better results when given enough time.

Table 2. Further Comparison

Instances	SBBO cost	BBOSLS(1) cost	BBOSLS(2) cost	BBOSLS(3) cost	Other Methods cost	method
Siemens1	47.019	2.285	2.226	2.4216	2.20	SA-DP
Siemens2	65.093	43.863	44.123	44.343	14.27	SA-DP
Siemens3	130.387	81.471	94.702	88.315	4.73	SA-DP
Siemens4	277.754	344.705	357.268	364.791	77.25	SA-DP
Swisscom	0	0	0	0	0	OA
K	39.739	24.732	44.277	11.35371	0.45	DTS-TA
Tiny	0	0	0	0	0	OA

4 Conclusion

In this paper, three hybrid BBO algorithms for the MI-FAP are proposed. The proposed method are evaluated on various benchmark problems, and compared to the standard version of BBO and GA. The experiments conducted show that the hybrid approaches perform better than the standard BBO, GA and the results are very encouraging. We plan to conduct other comparisons with others techniques and enhance different aspect of the marriage between the SLS/BBO to make it more efficient. It would be interesting to investigate if it is possible to obtain better results by increasing the search time for the proposed approaches. We plan to improve our work by parallelizing the steps in the proposed methods that can be launched in parallel.

References

1. Boughaci,D., Benhamou,B., H.Drias, "A memetic algorithm for the optimal winner determination problem", in *Soft Comput*, 13:905-917, 2009.
2. Hale, W .K, "Frequency assignment : Theory and applications", in *Proceedings of the IEEE 68(12)*, pp:1497-1514, 1980.
3. Kuurne, A., "Mobile Measurement based Frequency Planning in GSM Network. *PhD Thesis*", Helsinki University of Technology, 2001.
4. Kumar, G., Sachdeva, M. , "The use of artificial intelligence based techniques for intrusion detection", in *Artificial Intelligence Review*, vol. 34, no. 4, pp:369-387, 2010.
5. Luna, F. Blum, C., Alba, E., Nebro, A.J.,"ACO vs EAs for solving a real-world frequency assignment problem in GSM networks. in the *Proceedings of the genetic and Evolutionary Computation Conference (GECCO 2007)*, pp: 94-101, 2007.
6. Luna. F, Estebanez.C , L. Coromoto , Chaves-Gonzalez. JM, Nebro. AJ, Aler. R, Segura.C, Vega-Rodriguez. MA, Alba. E, Valls. JM, Miranda. G, Juan Antonio Gomez Pulido, "Optimization algorithms for large-scale real-world instances of the frequency assignment problem", it in *Soft Computing*. 15(5),pp: 975-990, 2011.
7. Mabed, H. , Caminada, A. , Hao , J.K., "Genetic tabu search for robust fixed channel assignment under dynamic traffic data", in *Coput Optim Appl* 50, pp:483-506 , 2011.
8. Montemanni,R . Moon, JNJ, . Smith D,H, "An Improved Tabu Search Algorithm for the fixed Spectrum Frequency Assignment Problem", in the *IEEE Transactions on Vehicular Technology*,, vol 52,issue 4, pp: 891- 901, 2003.
9. Simon, D., "Biogeography Based Optimization", in the *IEEE Transactions on Evolutionary Computation*,vol 12, pp: 702 - 713, 2008.
10. <http://fap.zib.de/problems/COST259/>

Various heuristics with cellular automaton to generate 2D shapes

Fazia Aïboud, Nathalie Grangeon and Sylvie Norre

LIMOS CNRS UMR 6158, 24 avenue des Landais.
BP10125. 63173 AUBIERE Cedex, France

{aïboud, grangeon, norre}@moniut.univ-bpclermont

Abstract: Cellular automata were introduced by Stanislas Ulam and John Von Neumann in an attempt to model natural physical and biological systems [1, 2]. They are discrete dynamic systems in space and time with simple local interactions but complex global behaviour [3]. They can be used to study complex dynamic systems such as self-organization phenomena [1], development of tumour [4], fire forest propagation [5], diffusion phenomena [6] and shapes generation.

The reason behind the popularity of cellular automata is due to their simplicity, and to the enormous potential they hold in modelling complex systems, in spite of their simplicity. Cellular automaton consists in a regular lattice of cells. The communication between cells is limited to local interaction. Each cell can take a state chosen among a finite set of states. This state can evolve over time depending on the states of its neighbour through a local evolutionary rule. The set of these local rules forms a transition function of the cellular automaton. The number of transition functions increases when the number of considered neighbouring cells and the number of states increase. This simple structure when iterated several times produces complex shapes displaying the potential to simulate different natural phenomena.

In order to obtain a given configuration of cellular automaton or behaviour after a given number of generations, it is very important to determine the initial configuration and/or the transition function allowing this configuration or this behaviour. This problem is known as the inverse problem.

The generation of shapes by cellular automaton is a particular case of the inverse problem of the cellular automaton. Two problems arise: the first one is how to choose a transition function and the second one is how often to apply this transition function (number of generations) allowing to the cellular automaton to evolve toward the desired shape.

The inverse problem of deducing the local rules from a given configuration or global behaviour is extremely hard [7, 8]. Several works in the literature propose to use the evolutionary computation techniques such as genetic algorithm to solve this inverse problem. The first major publication which makes genetic algorithms a popular tool for evolving cellular automata is due to [9]. In this work, the authors have applied a genetic algorithm to solve the computation task of the density classification problem and synchronisation of 1D cellular automaton.

In this paper, we consider the generation of 2D expected shapes. We propose a combination of search methods (heuristic and metaheuristic) with cellular automaton. The used metaheuristics are based on simulated annealing and iterated local search. The search method determines the transition function and the number of generations that allow to the cellular automaton to evolve and to give the nearest shape to the expected shape. Mutual information and number of identical cells are used as similarity criteria between two shapes. Different neighbourhood systems for the metaheuristic are proposed

Keywords: Binary cellular automaton, metaheuristics, optimization, neighbourhood system, mutual information.

References

- [1] Von Neumann, J. & Burks, A. W., 1966. *Theory of Self-Reproducing Automata*. Urbana and London: University of Illinois press.
- [2] Beyer, W., Sellers, P. & Waterman, M., 1985. Stanislaw M. Ulam's contributions to theoretical theory. *Letters in Mathematical Physics*, Volume 10, pp. 231-242.
- [3] Wolfram, S., 1984. Universality and complexity in cellular automata. *10D*, pp. 1-35.
- [4] Moreira, J. & Deutsch, A., 2002. Cellular automaton models of tumor development: a critical review. *Advances in Complex Systems*, Volume 2, pp. 247-267.
- [5] Karalylidis, I., 1999. Acceleration of cellular automata algorithms using genetic algorithms. *Advances in Engineering Software*, pp. 419-437.
- [6] Margolus, N., 1984. Physics-like models of computation. *Physica 10D*, pp. 81-95.
- [7] Kagaris, D. & Tragoudas, S., 2001. Von Neumann hybrid cellular automata for generating deterministic tests sequences. *ACMTrans. On Design Automation of Electronic Systems*, 6(3), pp. 308-321.
- [8] Ganguly, N. et al., 2003. *A survey on cellular automata*, : Technical Report, Centre for High Performance Computing, Dresden University of Technology.
- [9] Mitchell, M., Crutchfield, J. P. & Das, R., 1996. Evolving Cellular Automata with Genetic Algorithms: A review of recent work. *1st International Conference on Evolutionary Computation and its Applications. Russia*, pp. 1-15.

A general approach to solve decomposable optimization problems in multiagent simulations settings: application to tsunami evacuation

Van-Minh LE¹, Yann Chevalere² Ho Tuong Vinh¹, and Jean-Daniel Zucker³

¹ IFI, Equipe MSI; IRD, UMI 209 UMMISCO, Institut de la Francophonie pour l'Informatique Vietnam National University, Hanoi, Vietnam

² LIPN, CNRS UMR 7030, Université Paris Nord, 99 Avenue Jean-Baptiste Clement, 93430 Villetaneuse, France

³ UMI 209, UMMISCO, IRD France Nord, 93143, Bondy, France

Abstract. Nowadays, tsunami is becoming one of the most dangerous natural disaster for coastal regions. Along with the early warning system, evacuation is one of the first mitigation procedures to consider. The evacuation simulation then received a lot of studies in recent years in the domain of computer science. In fact, there are always the part of evacuees (e.g. the tourist) who lack information of the evacuation map, which motivates us to study the problem of optimizing of guidance sign placement for tsunami evacuation.

With regards to the problem of optimizing of guidance sign, most of the studies focus on two directions: minimize the average evacuation time and maximize the number of survivors. For the minimization of evacuation time, one linear programming approach was proposed which provided very fast optimization but lack of exactitude. Another approach is to build the agent-based simulation of evacuation and then optimize the input parameters (the position where to deploy the sign and the directions of each signs) to maximize the survivor. While this agent-based approach addresses exactly to the victim, it presents the limitation of computational requirement for exploring the huge parameter space. This paper presents an improvement of the agent-based approach that explores only the position of signs (rather than both position and direction), then optimizes the direction of each sign pointing to the nearest shelter, which reduces much parameter space. The direction optimization in the paper can be pre-computed and applied for all simulation during the evaluation of parameter exploration, which reduces much execution time.

Keywords: Linear Programming, Agent-based Simulation, Tsunami Evacuation, Sign Placement Optimization

1 Introduction

In recent years, whenever we talked about tsunami, we mentioned the terrible destruction and huge casualties (the tsunami from Indian Ocean in 2004 and the tsunami in Tohoku Japan 2011 [11]). Along with the early warning system, the

evacuation seems to be one of the most effective mitigation procedures. However, the evacuation of huge population in urgent situation is still complicated, because there are always some people who lack information about the evacuation map (e.g. the tourists). The deployment of guidance signs has been taken into account as a solution to minimize the number of victims in case of tsunami (the signs deployed in Palabuhanratu, Indonesia [12]).

With regards to the problem of optimizing of guidance sign, the objective function is normally the number of survivors who are people arriving the safe places (called shelters) within the predefined limited time and the parameter is the set of signs (including positions and direction). The optimization problem is then formalized as: $compute(\theta^*, \phi^*) = argmax f(\theta, \phi)$, in which θ^* is optimal position, and ϕ^* is optimal direction. However, the optimization problem on the two sets of variables is too heavy, especially when the evaluation of each solution (of both positions and directions) is produced by agent-based simulation which often requires much computational effort.

This paper proposes the approach that we make a very fast pre-computation which approximates one set of variables from the other and the optimization phase then focuses on only one set. The problem becomes: $precompute \phi = h(\theta)$; $compute(\theta^*, \phi^*) = argmax f(\theta, h(\theta))$. We also present the application of this approach to sign-placement optimization for tsunami evacuation in which the optimization phase only takes the positions of the signs as the parameters to explore, then approximates the direction from these positions by pointing to the nearest shelter and at last calculates the number of survivor by using agent-based simulation of evacuation. The direction of each sign is computed only one time for all the evaluation by simulation. This paper is organized as follow: Section 2 presents a brief state of the art on agent-based simulation of evacuation; In section 3, we present our approach to improve the optimization of sign placement using agent-based simulation and parameter exploration; We then present the implementation and verification of our proposition in section 4; In section 5, we present the conclusions and the perspectives.

2 Related works

2.1 Approach of Minimize Average Evacuation Time for Optimizing Sign Placement

One of the first ideas of minimizing the casualties is to minimize the evacuation time. An linear programming approach was proposed to Minimize the Average Evacuation Time (called MAET approach [7]) but the way to deploy the signs that minimizes Average Evacuation Time seems not always to lead to the maximizing the number of survivors, which leads us to find another approach that addresses exactly to maximization of survivors.

2.2 Approach of agent-based simulation

Another approach that addresses directly to this problem is the agent-based simulation. This optimal problem becomes building a model of evacuation simu-

lation (taking the number of survivors as objective function) and then exploring the parameter space (representing the road where to deploy the sign). That is why, in this paper, we only focus on the approaches that model pedestrian behaviors in the crowd and searching algorithm to optimize the sign placement.

Modeling the simulation and agent behaviors First, the traditional approach for modeling pedestrian behaviors in evacuation is modeling pedestrian movement on the grid. Each step of simulation, a pedestrian chooses one of the neighbor cells for his next location. Although this approach has received much success in simulating the evacuation from the building where the map is really small [9] [1], we argue that it would not be a good choice for the evacuation of tsunami because the evacuation map of a real city is much larger than that of a building.

Another approach proposed in [5] was to model pedestrian behaviors in tsunami evacuation. In this approach, the pedestrian movement was modeled as the transition through the petri-net. With this approach, modeler might add the rules inciting agents to go towards the less crowded road. While this approach provided the way to reduce casualties, we find that it is only suitable for the simulation in which all agents know the map. In fact, since agents normally "observe" what is happening locally, they can make their decision based on the local perception of the environment with the same rules you describe. That is why we do not use this approach in our work.

The Markov Chain Approach seems to be useful to model the simulation in which some people do not know all the map. While the authors in [4] used an evaluation function on every agent to make decision (which distinguishes the knowledge level of agent), the authors in [8] separated agents definitely into two types: one knows all the map, the other does not, which motivates us to use Markov Chain in this paper.

Exploring parameter space for optimization Once we use the Markov Chain approach to build the agent-based simulation of pedestrian evacuation, the only approach for optimizing of sign placement is the exploration of input parameter-space. For this exploration, in [10], the authors presented the comparison among the parameter-space exploiting methods (uniform random search, hill climbing and genetic algorithm). And among these methods, they argued to choose the genetic algorithm which inspired us to adapt to our proposed formulation. We then propose to use the genetic algorithm to exploit the parameter-space to optimize sign placement. For this particular problem, in [3], the authors presented the optimization of sign placement by using the genetic algorithm to explore the parameter space of simulation. In this study, they encoded chromosome from the list of edges on which the signs are deployed as the input, then evaluate the number of survivors (from the result of simulation) as the fitness. While this approach address directly to the number of survivors, we argue that the way to take the edges as the input parameter makes the space too large to explore. The exploration should be applied on the vertices rather than

the edges because in the graph representing a real map, the number of edges are much more than that of vertices.

3 Formalization of the optimization problem

In this paper, we formalize our problem as a typical optimization problem. We first build the simulation of evacuation in case of tsunami which takes the roads where the signs are deployed as the parameters and the number of survivors as the objective function. This simulation is then applied to the exploration algorithm as a evaluation function. The optimization problem becomes searching the set of roads which makes the simulation return the maximum number of survivors.

3.1 Modeling the simulation

First, our model focuses on 4 aspects of the simulation: environment, input parameter, agent behavior and outcome. First, the environment is modeled from GIS files which describing the roads and the high buildings which are used as shelters. The road GIS files are transformed into the graph $G = (V, E)$. The vertices which are nearest to the building (from building GIS files) are denoted as shelter vertices. Then, the input parameter is a list of pairs: one part of this pair representing the position of a sign (the vertex where the sign is deployed) and other describing the directions of this sign (the neighbor vertex to where the sign points). Next, the pedestrian behaviors in this model are: moving along the edge of the graph, choosing the next target when it arrives a junction (if there is a sign at the junction, the agent takes the direction of the sign as its next destination, otherwise it selects randomly one of the neighbor vertices). Finally, the outcome of the simulation is the number of survivors which are the number of agents arriving to any shelter vertex within the predefined limited time.

3.2 Genetic Algorithm on position and optimize direction to nearest shelter for Parameter Exploration of Agent-based Simulation

With regards to the parameter exploration to maximize the number of survivors, we propose to use genetic algorithm. The input parameter to simulation is then encoded as a chromosome and the number of survivors becomes fitness function. Besides, we propose to encode only positions of the signs (rather than both positions and directions). The direction of each sign is approximated by taking the shortest path to the nearest shelter. For an example that the position at vertex A (in the figure 1A) is a place to deploy a sign. The path (A, B, C, D, E) is the shortest path to the nearest shelter. Thus, for the sign situating at A, its direction should point to B. In our approach, the direction of each sign is independent from any other, we thus calculate the directions only one time

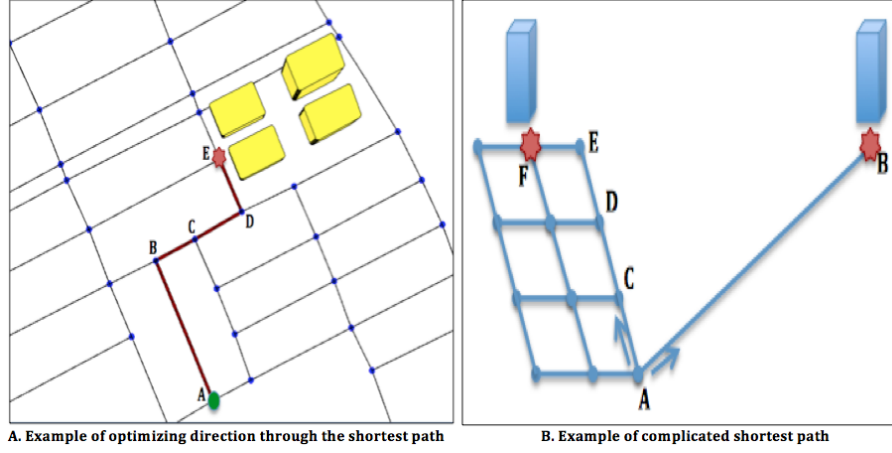


Fig. 1. Illustration of optimizing direction to the nearest shelter

for all the candidate vertices and then apply them to all the experiment of one simulation.

The shortest path for directing the sign in this proposition is just an approximated way, not yet the optimal approach. From observation of some cases of a very complicated shortest path, the evacuee often get lost. For an example of figure 1B, the path to shelter B is further but much simpler than the shortest path to the shelter F. For a sign situating at A, the shortest path approximation points the sign toward vertex C (the next vertex on the shortest path (A, C, D, E, F) to the nearest shelter), which makes the evacuees get lost at the corners C, D, E if they take a turn at one of these corners.

4 Implementation and evaluation

4.1 Description of evacuation model

In this section, we present the application case so that we can evaluate our proposition. In this case, we simulated the evacuation of pedestrians in the case of tsunami in Danang city in Vietnam. In our simulation we used the real map of this city and simulated 10000 pedestrians with the initial uniform distribution on over the junctions of the map.

The scenario of evacuation in this simulation is that: there were an earthquake of 8.0 Richter from Manila Trench which would cause the tsunami high up to 4m. The tsunami propagation time from the source region to Vietnamese coastal area were estimated about 2 hours [13]. According to limit-time scenario (described more detail in [6]), since the local government need time to consider before declaring evacuation alert and also for spreading information to people, they had 30 minutes to evacuate. They had to choose the high building for shelter (the

way that the people in the city move far from the beach were not recommended as the time of evacuation is limit).

In order to evaluate our proposition, we first built the Agent-based simulation of pedestrian evacuation by using GAMA [2]. The pedestrian behaviors in this simulation followed strictly our model described above and the outcome measure is the number of survivors, which means that the agent who arrived shelter within the 1800 seconds (30 minutes) is considered survivors. For the parameter exploration, we used genetic algorithm provided by ECJ framework [14] and fixed all other coefficients (e.g. selection, crossover, mutation, elitism) by default. This genetic algorithm takes the outcome of simulation (the percentage of survivors) as the fitness and the encode of chromosomes is separated into 2 implementations:

1. Genetic Algorithm on the position and direction to optimize the signs: this implementation encodes list of edges for the genetic algorithm and uses the Agent-base simulation to evaluate the fitness.
2. Genetic Algorithm on positions, optimization of direction of each sign pointing to the nearest shelter, the fitness is also evaluated by the simulation.

4.2 Evaluation

For the evaluation, we focus on 2 problems: the computational cost (whether our approach takes much execution time than the old one) and the result (whether ours provides the better result)

Execution time With regards to the execution time, we run 2 implementations with different number of agents. Each experiment, we run simulation 100 times and take the execution time as the measurement. The figure 2A shows that the execution time of both approaches are almost the same. The reason is that the optimizing direction phase is run only one time at the beginning of the evaluation and the direction optimal result is reused in all experiments.

Optimal result of sign placement to maximize the survivors For the result of sign placement to maximize the survivors in pedestrian evacuation, we run 2 implementation over 1000 generations (in genetic algorithm) with 300 signs for the simulation (which means that we have to place 300 signs in the city to maximize the percentage of survivors in the evacuation). The figure 2B shows that the Genetic Algorithm on position and optimize direction to nearest shelter produces the better result than Genetic Algorithm on both position and direction. Furthermore, the result from Genetic Algorithm on both position and direction is not even much better than the solution that the signs are placed randomly. From our observation, there are 2 reasons for this: first, the space of only position parameter is much smaller than that of both position and direction; second, the direction optimization avoid the circle problem while the stochastic crossover and mutation on both position and direction makes the solution fall

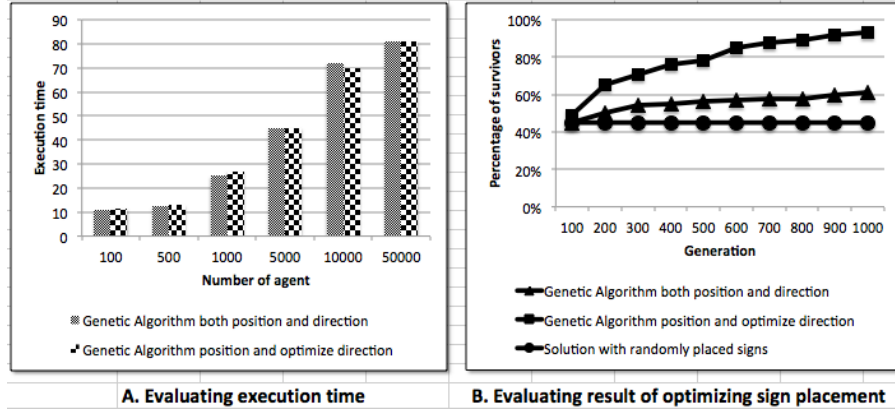


Fig. 2. Example of optimizing direction pointing to nearest shelter

into the circle problem. The circle problem here is that the signs point to each other and cause the circle. For example if a certain solution contains the 3 signs (A pointing to B, B pointing to C, C pointing to A), all evacuees arriving at one of these three vertices get lost.

5 Conclusion

In this paper, we proposed a pre-computation approach that accelerated the optimization. We then applied the approach to sign-placement optimization that used genetic algorithm on the sign position of sign and optimize the direction of each signs pointing to the nearest shelter and then evaluate the fitness by agent-based simulation. Our approach was seriously verified and validated. This can be transferred to many other agent-based simulation problems.

In the future work, we first focus on searching the solution for the complicated shortest path problem. Then we calibrate the genetic algorithm to improve the optimizing phase (which type of crossover, which type of mutation, which is the best value for each coefficient in genetic algorithm). Furthermore, we also intend to integrate the social factors into our evacuation model. These factors are the communication factor (which describes how an agent communicates with others for information), leader/follower factor (which means: the people often evacuate in group, the one who does not know the map can follow the other)

References

1. CHRISTENSEN, K.: Agent-Based Emergency Evacuation Simulation with Individuals with Disabilities in the Population. *Journal of Artificial Societies and Social Simulation* 11(3) (2008)

2. DROGOUL, A., AMOUROUX, E., CAILLOU, P., GAUDOU, B., GRIGNARD, A., MARILLEAU, N., TAILLANDIER, P., VAVASSEUR, M., VO, D.A., ZUCKER, J.D.: GAMA: Multi-level and Complex Environment for Agent-based Models and Simulations (demonstration). In: the 2013 international conference on Autonomous agents and multi-agent systems. pp. 1361–1362 (2013)
3. LE, V.M., CHEVALEYRE, Y., ZUCKER, J.D., HO, T.V.: Speeding up the evaluation of casualties in multi-agent simulations with linear programming, application to optimization of sign placement for tsunami evacuation. In: Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2013 IEEE RIVF International Conference (2013)
4. LIU, Y., OKADA, N., TAKEUCHI, Y.: Dynamic Route Decision Model-based Multi-agent Evacuation Simulation - Case Study of Nagata Ward, Kobe. *Journal of Natural Disaster Science* 28(2), 91–98 (2008)
5. MINAMOTO, T., NARIYUKI, Y., FUJIWARA, Y., MIKAMI, A.: Development Of Tsunami Refuge Petri-Net Simulation System Utilizable In Independence Disaster Prevention Organization. In: The 14th World Conference on Earthquake Engineering October 12-17, 2008, Beijing, China (2008)
6. NGUYEN, H.P., VU, H.P., PHAM, T.T.: Development of a tsunami evacuation plan for urban area of Nhatrang city using GIS. *Journal of Earth's sciences (in Vietnamese)* (2012)
7. NGUYEN, T.N.A., CHEVALEYRE, Y., ZUCKER, J.D.: Optimizing the Placement of Evacuation Signs on Road Network with Time and Casualties in case of a Tsunami. In: 20th IEEE International Conference on Collaboration Technologies and Infrastructures (WETICE 2011). pp. 394–396. No. i (2011)
8. NGUYEN, T.N.A., ZUCKER, J.D., NGUYEN, M.H., DROGOUL, A., NGUYEN, H.P.: Simulation of emergency evacuation of pedestrians along the road networks in Nhatrang city. In: Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012 IEEE RIVF International Conference. pp. 1–6 (2012)
9. SMITH, J.L., BROKOW, J.T.: Agent Based Simulation of Human Movements during Emergency Evacuations of Facilities. In: Structures Congress 2008. pp. 1–10. American Society of Civil Engineers, Reston, VA (Oct 2008)
10. STONEDAHL, F., WILENSKY, U.: Finding Forms of Flocking : Evolutionary Search in ABM Parameter-Spaces. In: MABS 2010. pp. 61–75 (2010)
11. SUPPASRI, A., IMAMURA, F., KOSHIMURA, S.: Tsunami Hazard and Casualty Estimation in a Coastal Area That Neighbors the Indian Ocean and South China Sea. *Journal of Earthquake and Tsunami* 06(02), 1–25 (Jun 2012)
12. TANIOKA, Y., LATIEF, H., SUNEDAR, H., GUSMAN, A.R., KOSHIMURA, S.: Tsunami Hazard Mitigation at Palabuhanratu, Indonesia. *Journal of Disaster Research* 7(1), 19–25 (2012)
13. VU, T.C., NGUYEN, D.X.: Tsunami risk along Vietnamese coast. *Journal of Water Resources and Environmental Engineering* (23), 24–33 (2008)
14. WILSON, G.C., MCINTYRE, A., HEYWOOD, M.I.: Resource review: Three open source systems for evolving programs-Lilgp, ECJ and grammatical evolution. *Genetic Programming and Evolvable Machines* 5(1), 103–105 (Mar 2004)

Intrusion Detection System In Mobile Ad Hoc Network Based On Mobile Agent

Y. EL Mourabit, A. Toumanari, A. Bouirden, H. Zougagh, R. Latif

Laboratory Signaux system & Informatique, Ibn Zohr University, ENSA Agadir, Agadir, Morocco.

Abstract. Mobile Ad Hoc Networks are a group of wireless computers, forming a communication network, that have no predetermined structure. It's highly vulnerable to attacks due to the open medium dynamically changing network topology, co-operative algorithm, lack of centralized. The fact that security is a critical problem. This work describes the proposal for an Intrusion Detection System architecture that uses multi-agent system, and a classification algorithm to detect the intrusion. Its an effective choice for many research and application areas due to several reasons, including improvements in latency, reducing network load and threat assessment. To respect the main primitives of a multi-agent system, we use the MadKit platform for implementation.

Keywords: MANET, Multi-Agent System, IDS, MadKit, Mobile Agent.

1 Introduction

Security in Mobile Ad-Hoc Network (MANET) is the most important concern for the basic functionality of network. Availability of network services, confidentiality and integrity of the data can be achieved by assuring that security issues have been met. MANET often suffer from security attacks because of its features like open medium, changing its topology dynamically, lack of central monitoring and management, cooperative algorithms and no clear defense mechanism [1].

Because the Mobile Ad Hoc Network has characteristics of wireless connected signal channels, autonomous mobile nodes, network topology in dynamic change and weak security authentication mechanisms, in addition, it is easier to suffer various security threats and attacks form passive eaves dropping to active impersonation, message playback, message falsification, and denial of service, etc, [2]. Therefore, the Intrusion Detection System (IDS) comes into the second firewall of network security solution. Intrusion detection is one of the key techniques behind protecting a network against intruders. An intrusion detection system tries to detect and alert on attempted intrusions into a system or network, where an intrusion is considered to be any unauthorized or unwanted activity on that system or network. An IDS is a defense system that detects hostile activities in a network and then tries to possibly prevent such activities that may compromise system security.

The detecting speed and high false positive rate of traditional intrusion detection system, the Intelligent and mobile characteristics of the agent are principals reasons to propose a new architecture of intrusion detection system based on mobile agent.

2 Related work

This section explores the work of researchers in the fields of mobile agent for intrusion detection system and highlighted the areas of potential scope for research: A distributed intrusion detection system using mobile agents (DIDMA) [11] performs decentralized data analysis using mobile agents that makes it more scalable. DIDMA uses platform independent components in contrary to platform specific security managers of CSM. The Autonomous Agents for Intrusion Detection (AAFID) project [3] makes use of multiple layers of agents organized in a hierarchical structure with each layer performing a set of intrusion detection tasks. A proposed efficient anomaly intrusion detection system in Ad-hoc by mobile agents [4] which uses the data mining algorithm to detect the attacks exploited by the intruders. Mobile agent based intrusion detection system for MANET [5] proposed by Yinan Li which uses the clustering and joint detection technique to identify the intruders.

Literature review [6] brought up the fact that although many attempts have been made to provide security in MASs (Mobile Agent Systems) communication and establishing trust among the agents, many rigid technologies developed to support security; but as the wheel of the technology spins every time, so the area always needs further refined researches in every approach we take.

3 Survey of Mobile Agent and IDS

3.1 Mobile Agent

Mobile Agents are the programs that move between computers or nodes of network, autonomously trying to fulfill some specific goals given by users. Agents are different from other applications in that they are goal-oriented: they represent users and act on their behalf to achieve some set goals in an autonomous manner i.e. they control themselves, as in the decision where and when they will move to the next computer or node. Mobile agent provides a viable means of analysis efficiently and effectively. Mobile agent neither brings new method to detect for IDS nor increases detection speed for some kind of attracting. Nevertheless, it improves the design, construct, and execute of IDS obviously. Mobile agents offer several potential advantages that may overcome limitations that exist in static, centralized components:

Reducing Network Load: Instead of sending huge amount of data to the data processing unit, it might be simpler to move the processing algorithm (i.e. agent) to the data.

Overcoming Network Latency: When agents operate directly on the host where an action has to be initiated, they can respond faster than the tree based systems that have to communicate with a central coordinator located elsewhere on the network.

Autonomous Execution: When portions of the tree based systems get destroyed or separated, it is important for the other components to remain functional. Independent mobile agents can still act and do useful work when their creating platform is unreachable which increases the fault-tolerance of the overall system.

Heterogeneous Environment: The agent platform allows agents to travel in a heterogeneous environment and inserts an OS independent layer.

Dynamic Adoption: The mobility of the agents can be used to reconfigure the system at run-time by having special agents move to a location where an attack currently takes place to collect additional data.

Scalability: When distributed mobile agents replace a central processing unit, the computational load is divided between different machines and the network load is reduced. This enhances scalability and additionally supports fault-resistant behavior [7].

3.2 Intrusion Detection System

IDSs are hardware and software systems that monitor events occurred on computers and computer networks in order to analyze security problems. IDS and firewalls have become key components in ensuring the safety of network systems. Intrusions and invasions inside computer networks are called as attacks and these attacks threaten the security of networks by violating privacy, integrity and accessibility mechanisms. Attacks can be originated from users who login to the computer using Internet trying to gain administrator rights and other users who misuse the rights they have. IDSs automate monitoring and analyzing the attacks [8].

In general, the IDSs are composed of four components (sensors, analyzers, database and response units) and are responsible for activities such as monitoring the users and systems activities, auditing systems configuration, accessing data files, recognizing known attacks, identifying odd activities, auditing data manipulation, tagging normal activities, error correction and storing information concerning invaders [9].

There are four basic techniques used to detect intruders: Anomaly detection, misuse detection (signature detection), target monitoring, stealth Probes.

4 Proposed Work

Many IDS for ad hoc network have proposed. Some of them have critical for certain scenarios. Some of them are used with collaboration of routing protocols. Here we propose a new Intrusion Detection System based on the mobile agent which employ statistical classification algorithms to order to perform intrusion detection in MANETs. Such algorithms have the advantages that they are largely automated, that they can be quite accurate, and that they are rooted in statistics. For that reason, they are prime candidates for use in cost-sensitive classification problems. After training, they can be used for detection with arbitrary cost matrices. They have extended applications including intrusion detection in wired networks [10]. In standard classification problems the classification decision is selected in order to minimise the probability of error. However, to detect the intrusion, we perform a thorough comparison of three well-known algorithms for the detection intrusion in Mobile Ad hoc Networks (MANETs), (K-Means, Naive Bayes, SVM) j:mac.

4.1 System architecture:

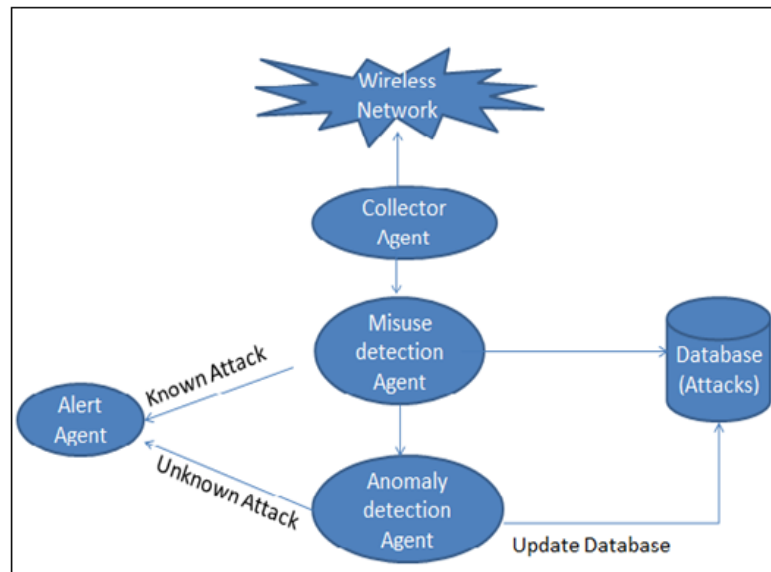


Fig. 1. System Architecture

Collector Agent: Collector Agent is the first agent to work in the system, it collects the data from the wireless environment, store those data in a file, which is given as an input to the misuse detection agent.

Misuse Detection Agent: Misuse Detection Agent analyses the data captured by the collector agent. It detect the known attacks in network by pattern matching algorithm, it reports to alert agent if there is a similarity between the collected packets and attack signatures in the database, if not, those data are given as an input to the anomaly detection agent.

Anomaly Detection Agent: Anomaly Detection Agent is used to detect the new or unknown attacks by using the classification algorithm SVM. If the incoming data is detected as attack, it reports to alert agent about the attack, and updates the detected attack in the database.

Alert Agent : The alert agent is used to alert the system if an intrusion occurs in the network. it alerts the system based on the output of the misuse and anomaly detection agent.

5 Simulation and Plateform of implementation

5.1 Simulation:

Firstly, we improve the reason of choosing SVM as a Classification algorithm to detect intrusion in anomaly detection agent. The dataset used in this experiment is the NSL-KDD dataset [13] which is a new dataset for the evaluation of researches in network intrusion detection system. It consists of selected records of the complete KDD 99 dataset [13]. NSL-KDD dataset solve the issues of KDD 99 benchmark and connection record contains 41 features. We use the Weka [15] tool to implement algorithms and to perform the rate classification. A simple

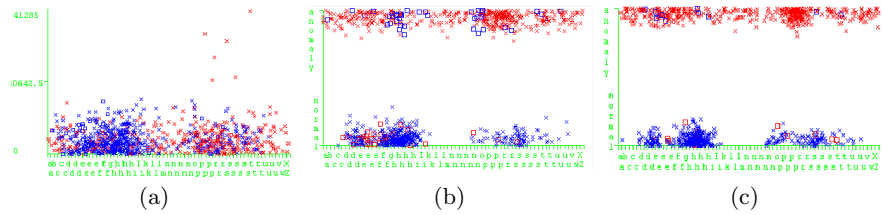


Fig. 2. K-Means cluster assignments(a), Naive-Bayes Classifier errors(b), SVM Classifier errors(c)

way to perform intrusion detection is to use a classifier in order to decide whether

some observed traffic data is normal or anormal (in the graphs above normal in blue color and anormal in red color). The classification objective is to minimise the probability of error. According to the results obtained we improve that

Table 1. Rate of classification

Algorithm Correctly Classified Instances	
K-Means	91,5%
Naive Bayes	92,2%
SVM	97,4%

the SVM classifier algorithm is more efficient than K-Means and Naive Bayes classifier with a classification rate reaching 97.4%.

5.2 Platforme of implementation

The first major task was to choose the right technology. Many of the technologies were checked like Agent Development Kit (ADK), JADE [14] and Aglet Software Development Kit (IBM).

These are well-known available platforms. The above mentioned technologies provide a platform for Agent development. ADK is still in its development phase and has lot of problems regarding agent movement which is the core essence of our research. JADE is FIPA [14] compliant Agent development framework thats why it does not provide more standards for agents mobility. Though most of the platforms have their own features and limitations but keeping in view the key property of our research, i.e. mobility, we chose MadKit platform. It provided us with the control we needed.

MadKit is a scalable and modular multi-agent platform, written in the Java language. It allows the creation of SMA based on the relational model Agent, Group, Role. MadKit leverages the Object Oriented Programming: the madKit features are contained in the kernel MadKit. This core is a set of classes for the user to design a basic way of SMA simple, but also, through inheritance, to design and add new features that will be compatible with those provided base. One of the biggest advantages is that MadKit, because it defines a basic structure for the representation of an agent, of a group, of a message. It is relatively easy to communicate with agents designed by programmers different, even for different projects.

The scenario for which our approach was tested is the game between two teams because it is considered a typical example of intrusion detection. We consider a game with two teams, on one side the intrusion detection system (IDS) and the other side the intruder. IDS seeks to flush out the attacks intruders protect the network. Furthermore, the intruder tries to reach its target and fulfill

his plan. In intrusion detection, attack consists of several packages. The intruder reaches its target when all the packets arrive to target. But, if the IDS arrives to an intercepting many of these packages, the attack is then flushed out. In reality the two adversaries adopt strategies at different levels and in different ways to achieve their goals.

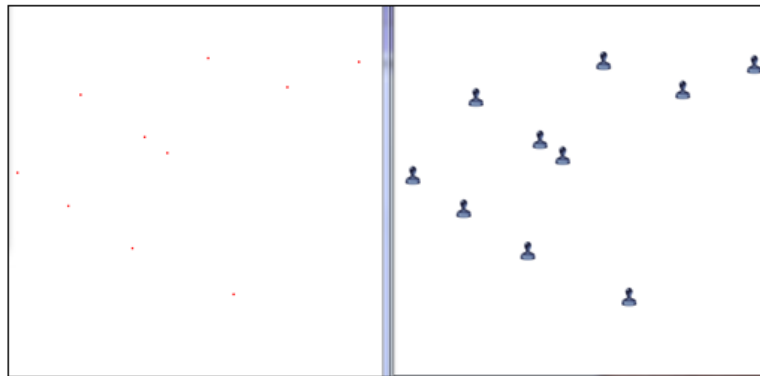


Fig. 3. Representation of mobile agent

6 Conclusion and future work

The proposed IDS exploits the benefits of employing mobile agents such as reduced ad hoc network bandwidth usage, increased scalability and flexibility, and ability to operate in heterogeneous environments. Here we are in position to say that mobile agents do provide a viable means of performing ad hoc network security analysis as well as some other complex tasks. As opportunities for future work, it could be identified: the deployment of a more complex detection, with mobile agents, using statistical anomalies detection identified by mobile agents and enabling the creation of attack signatures, the development of more complex detection ontology, with more parameters to characterize the attacks; the study of the impact of the use of the proposed architecture in ad hoc network traffic, and the implementation and testing of the architecture with a redundant and fault-tolerant main container.

References

1. ZUBAIR, DR., SHAIKH, A.: A platform independent approach for Mobile Agents to monitor Network Vulnerabilities. Proceedings of the 5th WSEAS Int. Conf. on APPLIED INFORMATICS and COMMUNICATIONS, Malta,(2005).

2. Suryawanshi, G.R., Jondhale, S.D., Korde S.K., Ghorpade P.P., Bendre M.R.: Mobile Agent for Distributed Intrusion detection System in Distributed System. (IJCTEE) Volume 1, Issue 3.
3. Balasubramaniyan, J., Fernandez, J. O. G., Isacoff, D., Spafford E. H., Zamboni, D.: An Architecture for Intrusion Detection using Autonomous Agents Technical report no. TR 98-05, Purdue University, USA, 1998.
4. Yinan, Li., Zhihong, Q.: Mobile agents based intrusion detection system for mobile Ad-hoc network in International Conference on Innovative Computing and Communication. pp: 145-148, March 2010.
5. Jaisankar, N., Saravanan, R., Duraisamy, K.: Intelligent intrusion detection system framework using mobile agents in International Journal of Network Security and its Applications. Vol 1, No 2, July 2009.
6. William M. Farmer, Joshua D. Guttman, Vipin Swarup Security for Mobile Agents Issues and Requirements, In Proceedings of the 19th National Information Systems Security Conference, Vol. 2, pp. 591-597. National Institute of Standards and Technology, Baltimore, Maryland, October 1996.
7. Blakley, B.: The Emperors Old Armor Proc. New Security Paradigms Wksp., 1996.
8. Bace, R. G.: Intrusion detection. SAMS, 2000.
9. Farid, D., Rahman, M.: Anomaly network intrusion detection based on improved self adaptive bayesian algorithm. Journal of computers, Vol. 5(1), Jan 2010, pp. 2331, doi: 10.4304/jcp.5.1.23-31.
10. Lee, W., Stolfo, S.J., Mok, K.W.: A data mining framework for building intrusion detection models Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 1999, pp.120132.
11. Kannadiga, P., Zulkernine, M.: DIDMA: a distributed intrusion detection system using mobile agents 06/2005; DOI:10.1109/SNPD-SAWN.2005.31 ISBN: 0-7695-2294-7 In proceeding of: Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005.
12. MacQueen, J.: Some methods for classification and analysis of multivariate observations Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, 1967, pp. 281.297.
13. NSL-KDD Data set for Network-based Intrusion Detection Systems. Available at: <http://nsl.cs.unb.ca/NSL-KDD/> [accessed: December 15, 2010].
14. Mosqueira-Rey, E., Alonso-Betanzos, A., Guijarro-Berdinas, B., Alonso-Rios, D., Lago-Pineiro, J.: A Snort-based agent for a JADE multi-agent intrusion detection system Int. J. of Intelligent Information and Database Systems, 2009 Vol.3, No.1, pp.107 121
15. WEKA (2008): Data Mining Machine Learning Software [Available Online] <http://www.cs.waikato.ac.nz/ml/weka/>.

The use of ontology in semantic analysis of the published learners messages for adaptability

Samia Ait Adda¹ and Amar Balla²

National High School of Computer, Ouad Smar, Algiers, Algeria
s.ait.adda¹, a.balla²{@esi.dz}

Abstract. Communication tools for online learning environments are ways that let learners to exchange messages between them and with their teachers. It is also a way to interpret their social behavior patterns and their learning styles. In this paper, we are interested in the semantic analysis of the contents messages published by learners by use of domain ontology. The purpose of this analysis is to identify the domain concepts that are most published and shared by learners and to keep them into the learners model as concepts not well mastered. We hypothesize that all concepts edited and exchanged over email, chat and especially in discussion forums can be considered as knowledge poorly or badly acquired by learners and deserve thus more attention and consideration both by the tutor for the pedagogical monitoring of learners on these concepts and from designer of course, to restructure and more enrich the educational content which articulates these concepts identified beforehand in this analysis.

Keywords: e-Learning, Communications tools, Domain ontology, Semantic indexing, Learner model, Adaptability.

1 Introduction

Our work is placed in the field of the Interactive Educational System; we set us therefore in a context of an online learning. Hence, learners work remotely on platforms which allow them to attend course, make tests and exercises or discuss by means of communication tools that are always integrated on these platforms.

The progressive integration of these communication tools offers opportunities for learners geographically dispersed to exchange without any time constraint and allows considering a new forms of social interaction between learners and between learners and teachers[1]. With this simplified sharing of information, a new form of interaction have emerged, and new skills have been developed both in social, cognitive, or meta-cognitive[2].

These communication tools are generally places of meetings and discussing for learners who are often in difficulties on some concepts of the taught domain, which may be weakly assimilated. If we take the example of a learner who publishes a question on a forum, this fact can interpret an obstacle of this learner on the concepts contained in the content of this question. As well as the messages exchanged between learners by email or chat.

We aim in this article to analyze the content of messages discussed by the learners to identify the most exchanged concepts and be able to help learners with their needs. We introduce among other things, to do this, ontology of teaching domain to correlate information between the different contents of messages published by the learner and those of the studied course. The goal that we want to achieve through this study is essentially enriching and updating the learner profile, by marking the domain concepts that pose a problem for him and that are detected due to its posted messages. Recognize the problematic concepts, of the learners participating in the learning process, can help the designer of the course to review the course content on these concepts and adapt it to the learners' level, and the tutor to assist and support learners who are already published these concepts. Hence, here we are talking about the adaptability of learning.

The paper will be structured as follows, in the first section we present some research that are carried on the analysis of communication tools and their users, then we are interested in some semantic web tools, such as ontologies and the possibility, thanks to these tools to exploit the contents of messages to detect concepts badly acquired by the learner, further we detail and explain our approach. Finally we present some results which we have achieved through an implementation of our approach on a considered domain.

2 Related Work

In several studies, the ability of message to structure the written exchanges may offer an interesting educational trail for helping learners to succeed. Some works [3] analyze the external factors of messages, these analyzes attempt to account for what is played on forums from readily observable indicators such as the number of posts by each user or group, the number of responses, the length of discussions, the number of learners participating in discussion, the average duration of a session on the forum ... etc.

While others were focused to analyze the internal ones. Indeed various researches [4, 5] have conducted a study to evaluate the use of data and text mining to analyze the learners' discussions, they propose systems for classifying textual contribution, such as: topics that come up in the debates, announcements, questions, students initiatives or answers, conflicts ... etc. Another type of frameworks [6, 7] have been proposed for characterizing and analyzing discussions for the classification and structuring, based on the Semantic Analysis which includes the tools of Web Semantic, like ontology, to organize the vocabulary and refine the analysis.

These studied works aim to show the cognitive, social and semantic advantage of published messages. Our approach is inscribed in the same perspective. On the other hand, we propose a method which gives a semantic meaning to learners messages with an ultimate goal of detecting concepts of the studied domain that are poorly or badly learned by the learner and keep them into his profile for adaptability. To do this, the concepts of the studied domain are modeled through ontology of the taught course, as we describe it below.

3 Domain Ontology

The Semantic Web [8] is an understandable and navigable space by both human and software agents. It introduces an additional meaning to the navigational data of the classical web, based on a formal ontology and controlled vocabularies through semantic links. In standpoint of e-learning, it can help learners to locate, access, querying, processing and evaluating learning resources across distributed heterogeneous network, or assist teachers in creating, using, locating, or the sharing and exchanging learning objects. Ontology [9] includes a set of terms, knowledge, including vocabulary, semantic relations, and a number of logic-inference rules for some particular domain. The ontology applied to Web creates thus the Semantic Web [10]. Ontologies [11] facilitate the sharing and reuse of knowledge, i.e. a common understanding of diverse content by persons and machines.

The use of ontology in our case consists in the conceptual indexing of the edited messages to facilitate their identification and semantic search by the learner since they become learning knowledges basis available for consulting [12]. On top of that indexing, the most edited domain concepts will thus detected. This ontology also represents the structure of the learners model, since it is part of the domain model, i.e. the domain ontology in our case.

In our case of study, we consider that ontology is composed of a set of concepts and relations between these concepts. A unique identifier is assigned for each concept, these concepts are labeled with one or several terms. Expressly an ontology O is defined as follows: $O=\{C, R, Vo\}$

- C : set of domain ontology concepts
- R : set of relations between the ontology concepts,
- Vo : is the ontology vocabulary which is composed with terms (mono or composed words) corresponding to the domain ontology concept.

4 The proposed approach

To detect knowledge of domain supposed poorly assimilated by learners, we propose an approach that consists in constructing a text document relative to the messages published by the learners in their email, chat and discussion forums.

These documents are taken from the database of the platform and then indexed according to the concepts of the taught course through the domain ontology (semantic indexing). The most edited concepts by a learner or group of learners are then highlighted. Our goal is threefold. On the one hand we try to index the messages to facilitate their research and consultation, on the other hand to detect the most edited concepts of domain by learners, and finally we want to identify learners who have used these concepts via the communication tools to make inquiries about some domain concepts.

The approach that we propose is divided into three basic processes: (1) Building of the messages corpus, (2) Semantic Indexing and (3) the management and processing of results, this is what will be detailed in the following of this paper.

4.1 Building of the edited messages corpus

To perform indexing, we need textual content messages. It just consists to access to the database of the platform to get messages exchanged by all learners from the corresponding tables. The result will be in the form of text documents, thus closing the content of message, paths to the attachments if exists, information about the sender and receivers of the message and the time of dispatching. These documents will be saved in a repository to constitute a local corpus of documents.

Below, a diagram summarizes the structure of the message in learning platform, that we will indexed with the concept of the studied domain via an ontology of domain, which is the purpose of this work.

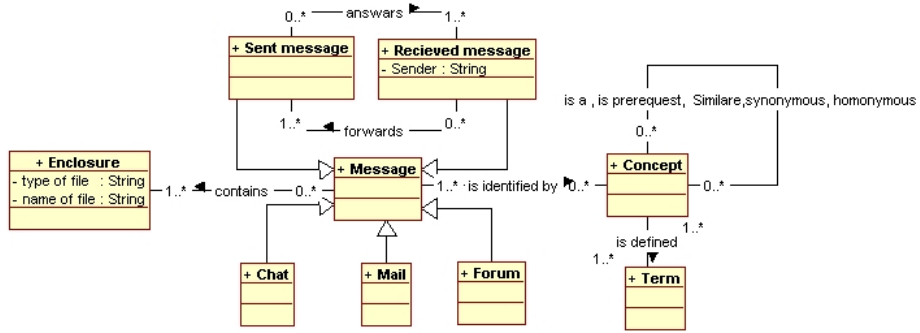


Fig. 1. The ontology model of message

4.2 The Semantic indexing

We propose to use domain ontology to build the semantic index of documents. The process of document indexing is handled through three main steps: (1) Identifying ontology concepts, (2) Assigning concepts to document terms and (3) Weighting concepts. In the following, we present these steps.

Concept Identification. The purpose of this step is to identify ontology concepts that correspond to document words. Concept identification [13] is based on the overlap of the local context of the analyzed word with every corresponding domain ontology entry. Concepts are referred in the text documents with simple or compound words (term). The concept identification algorithm is given in figure 2.

In the ontology, a set of terms is used for labeling concepts and relationships between concepts. That set forms the vocabulary of the ontology. To respond nevertheless in case if the processed term is ambiguous, a disambiguation step is so necessary.

Term Disambiguation. Each term ti (simple or compound words) in document may be associated with a number of related possible ontology concepts.

```

Input: document D.
Output: Vector of all ontology concepts belonging to terms in document D.
Procedure
Let wi be the next word to analyze in the document d. We define the context senti which is the sentence
in document D that contains the word occurrence wi being analyzed.
Compute  $V_i = \{C_1, C_2, \dots, C_n\}$  the of ontology entries containing wi.
Each  $C_j \in V_i$  is represented by a multiword or mono-word term.
Rank concepts  $C_j$  in set  $V_i$  in where:  $|C(1)| > |C(2)| > \dots > |C(n)|$  //  $|$  denotes the concept length,
in terms of the number of words in the corresponding terms.
For each element  $C_j$  in  $V_i$  do
    Get common words between senti and representative term of  $C_j$ , which is the intersection
     $N = \cap(\text{sent}_i, C_j)$ 
    If  $|N| < |C_j|$  then The concept-sense is not within the context.
EndIf
If  $|N| = |C_j|$  then The concept-sense  $C_j$  is within the context senti.
    Add  $C_j$  to the set of vectors' element (index) associated to the document D.
EndIf
EndFor

```

Fig. 2. The algorithm of terms Mapping into Concepts

Thus we distinguish the situation of semantic or polysemy ambiguity. That set forms the vocabulary of the the situation of semantic or polysemous ambiguity. For example, the term "table" has a three meaning in PHP ontology: (1) table of data structure, (2) table in database and table in the html structure. It can refer to three different concepts. In this case we proceed as follows, for an ambiguous term *ti* in the document, we seek a label of a concept *Ck* linked in the ontology with a concept *Ci* which is indicated by the ambiguous term *ti*. If *Ck* exists, *Ci* is taken as the concept designated by the term *ti*.

Concept Weighting . The extracted concepts are weighted according to a method more general than $tf * idf$ named $Cfc * idf$ (concept-frequency-inversed document frequency). In this method each extracted term represents necessarily a concept of the ontology since we used ontology to identify them. For a concept C its frequency in a document depends on the frequency of the term itself [14]. It is calculated as follows:

$$idf_c = \log \frac{n}{f_c} + 1 \quad (1)$$

$$Cfc = \sum_{tm \in t(c)} tf_{tm} \quad (2)$$

Where: $t(c)$ is the set of terms corresponding to different concept C and tf_{tm} is the frequency of term $t(c)$ in document *i*. The weight of each concept in a document *d* is so calculated as follows:

$$CfIdf = Cfc \times idf_c \quad (3)$$

5 Semantic representation of learner knowledge

Each indexed document is represented by a vector of weighted key concepts. For this purpose, all documents that constitute the corpus C will be represented by an occurrence matrix of document and concept. We will distinguish two kinds of corpus, a first corpus $C_{learner}$ consists of each message published by a learner, and a second C_{Grp} represents the sum of all messages edited by all learners who participated in the learning process.

Thanks to the first corpus, we can specify the knowledge which poses a problem to the learner. Since the occurrence matrix, we naturally recognize the most published concepts by the learner through adding instances of the same matrix row; the result is a vector $V_{learner}$ containing the concepts occurrences in the corpus $C_{learner}$.

$$V_{learner} = (Wc1, Wc2 \dots Wcn)$$

Concepts with a high weight by estimating a threshold α , that we will fixed by experimentation, will be reviewed as problematic domain concepts for a learner. Therefore, the tutor may intervene to help the learner on these concepts.

In fact, we distinguish several ways to represent the learners knowledge; the largest used method is *the Overlay model* [15]. This model represents the learners knowledge as a subset of the domain model, which reflects the expert-level knowledge of the subject. The domain model is presented in our case of study as domain ontology.

Hence, we propose an overlay knowledge model with *two layers* (2 levels). The first layer concerns *assessment* and contains the mark obtained by the learner in the test on the concept. As for the second layer, it stores the weight of the edited concept on *the communication tools* using the equation(3). Consequently, the learning model in the proposed system has been shown in an ontological form since using the domain ontology to represent the domain model.

For the general corpus C_{Grp} we proceed with the same process as a learner, so the result will be a general vector of concepts occurrence in the corpus C_{Grp} . Similarly, the concepts greater than a threshold β , which we also determine the value by evaluation, will be considered as wrong developed concepts in the course. To this end, the designer of course can review the content of resources that explain these identified concepts, and further enrich its course on these concepts.

6 Experimentation

6.1 The Test Collection

For our experiments, we have proceeded to test on group of computer science students in the second years, with the number of 27. We have proposed to them PHP course, shown in *eFAD* (www.ufc.efad.dz) platform and modeled with the ontology of SKOS format¹ (Simple Knowledge Organization System). The experiments were established in three sessions of one hour. The *PHP* course is mainly composed of 8 top concepts and 49 sub-concepts. To consolidate our experiment, we have conceived a questionnaire paper which we have distributed to students, asking them to place concepts which pose problem to them. At the end of the test, a written assessment was performed for all learners.

¹ <http://www.w3.org/2006/07/SWD/SKOS/reference/20090315/implementation.html>

6.2 Evaluation of results

The result of the experimentation consists of 27 corpus of each learner plus the general corpus. Therefore, we constituted a number of 105 textual documents of the messages extracted from the platform database. The following diagram shows the score characterizing the main domain concepts for each learner :

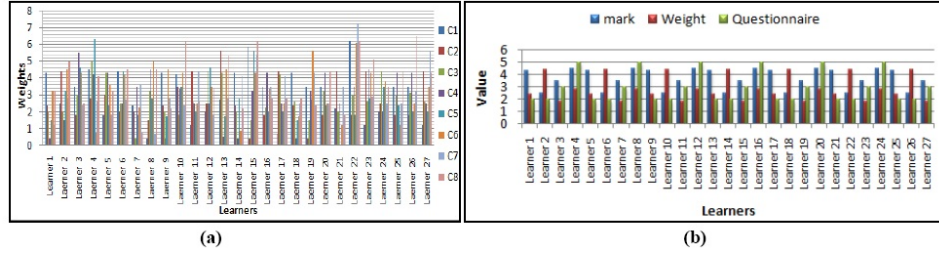


Fig. 3. The editing weights of the main domain concepts

The challenge of this test was to find the concepts insufficiently mastered by each learner. That is to say; the threshold α , that we set to evaluate the most concepts which posed a problem for student j , is estimated by the median of the weights of concepts edited by this learner. This threshold is different from learner to learner; accordingly we have counted 27 values of this threshold (figure 3.a). Indeed, we found that the concepts C4 and C6 have posed a problem for some students who are recognized by the following process.

As that is signaled, a written assessment was performed for each learner on each concept of the domain as well as a questionnaire which we have asked them to indicate the concepts not mastered. Therefore, the diagram in the figure 3.b shows a comparison of different results obtained from the assessment, the weight of edited concepts and questionnaire responses. As for the threshold β , which we considered to estimate the concepts which are badly defined in course, it is determined by comparing the result with that obtained through the written assessment and questionnaire responses, the value is fixed at 0.24. As a result, we detect that 5 learners have problems with some concepts of the domain (learners 2, 10, 14, 18, 22).

7 Conclusion

The educational adaptivity systems provide good support for learners on their individual characteristics. It can also provide information on the needs and deficiencies of these learners, either for the tutor or designer of the course, even for evaluation, monitoring and customizing the process and strategy of learning. In fact, the learner model must be developed for each student, containing information about the history of social interactions, objectives and knowledge badly acquired. In this article, we have highlighted the need to analyze the messages edited by learners during the learning sessions and we have proposed an approach for semantic analysis that we have presented and explained which permits to detect domain concepts that were difficult for learners, by

comparing the content of their messages with a domain ontology of the studied course. An experiment was carried out on a group of students taking a PHP course, and has enabled us to validate the proposed approach and to set some parameters. This result needs to be further refined by additional tests, which we are currently conducting.

References

- [1] Develotte,C., Mangenot,F., Nissen, E.: *Actes du colloque Epal 2009 Echanger pour apprendre en ligne : conception, instrumentation, interactions, multi-modalit*, universit Stendhal- Grenoble 3,5-7 juin (2009)
- [2] Baker, M.: *Forms of cooperation in dyadic problem-solving*. In: Revue d'Intelligence Artificielle, 16 (4), pp.587-620 (2002)
- [3] Huynh Kim Bang, B., Bruillard ,E.: *Vers une nouvelle interface de lecture pour des forums de discussion ddis des laborations collectives*. In actes de H2PTM05, Paris: Lavoisier, p.43-56 (2005)
- [4] Azevedo, B.F.T., Reategui,E. B., Behar,P.A. : *Qualitative Analysis of Discussion Forums*. In: IADIS International Conference on e-Learning,Freiburg, Alemanha, pp. 251-258 (2010)
- [5] Breno, F., Terra A. P., Alejandra B., Eliseo B. R.: *Automatic analysis of messages in discussion forums*. In: 14th International Conference on Interactive Collaborative Learning (ICL2011) 2123 , Piesany, Slovakia (2011)
- [6] Gloor, P.A., Zhao, Y.: *Analyzing actors and their discussion topics by semantic social network analysis*. In: Conference on Information Visualization. pp. 130135 (2006)
- [7] Leprovost, D., Abrouk, L., Gross-Amblard, D.: *Discovering implicit communities in web forums through ontologies*. Web Intelligence and Agent Systems: An International Journal 10, 93103 (2011)
- [8] Dicheva, D.: *Ontologies and Semantic Web for E-Learning*, In : Handbook on Information Technologies for Education and Training, 978-3-540-74155-8, Springer Berlin Heidelberg (2008)
- [9] Siti, U.,Rohiza, A. , Shakirah, M.: *Ontology of Programming Resources for Semantic Searching of Programming Related Materials on the Web*. 978-1-4244-6716-7110, IEEE (2010)
- [10] Berners-Lee, T., Hendler, J., Lassila, O.: *The semantic web, A new form of web content that is meaningful to computers will unleash a revolution of new possibilities*. Scientific American, pages 3543 (2001)
- [11] Zschocke,T., DeLeon, J.: *Towards an Ontology for the Description of Learning Resources on Disaster Risk Reduction*. In: CCIS 111, pp. 6074, Springer-Verlag Berlin Heidelberg (2010)
- [12] Refhay, C.,Greffier, F.: *Les chos du forum de discussion en FAD*. In JOCAIR 2006, Premires journées Communication et Apprentissage instruments en rseau, Amiens, France, 130-144 (2006)
- [13] Salah,T. B.,Khaled, M. F. , Naveed, A.: *Personalized Semantic Retrieval and Summarization of Web Based Documents* . In: (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.1 (2013)
- [14] Dragoni,M. ,Pereira, C., Tettamanzi, A.: *An Ontological Representation of Documents and Queries for Information Retrieval Systems*. In IEA/AIE 2010, pp. 555564, Springer-Verlag Berlin Heidelberg (2010)
- [15] Carr,B. ,Goldstein I.: *Overlays:A theory of modeling for computer-aided Instruction*. Massachusetts Institute of Technology, Cambridge, Massachusetts (1977)

Multiple Mobile Target Tracking in Wireless Sensor Networks

Charly Lersteau, Marc Sevaux, and André Rossi

Université de Bretagne-Sud, Lab-STICC
Centre de recherche, BP 92116
56321 Lorient cedex, France

Abstract. An *object tracking sensor network* (OTSN) is made of m *static* wireless sensors scattered throughout a geographical area for tracking n *mobile targets*. Assuming that sensors have non-rechargeable batteries, one of the most critical aspects of OTSN is energy consumption. In this paper, we propose linear programming models which handle two missions : *monitoring* and *reporting* data to a *base station*, and two distinct problems : *minimize energy consumption* and *maximize network lifetime*. We suppose that trajectories of targets are known and targets should be monitored by sensors. To reach our goals, we *schedule* the *active* and *sleep* states of the sensors and *route* the data to a base station while keeping track of the targets. To solve our problems, we process a *temporal discretization* according to the intersection points between the trajectories and the sensing ranges of the sensors. The obtained sets of sensors for each time window help us to create *linear programming* models. These basic problems offer perspectives in performance evaluation of energy-conservation protocols and distributed algorithms in wireless sensor networks.

Keywords: multiple target tracking, wireless sensor networks, lifetime maximization, energy consumption minimization

1 Introduction

During the last decade, Wireless Sensor Networks (WSN) have become more and more affordable and the number of application areas has increased. Sensor networks find their applications in battlefield or traffic surveillance, wildlife studies or healthcare [5, 7]. This paper focuses on target tracking using WSNs. Our problem is to cover *moving targets* using randomly deployed sensors. As the sensors used are often low-cost, a critical aspect of their deployment is battery limitation.

Two scenarios might occur: 1. battery capacity of sensors is large enough to ensure monitoring of all the targets during the whole time horizon ; 2. the targets cannot be monitored until the end of the time horizon. In the first case, our goal is to minimize the energy consumption, whereas in the second case, it is to maximize the network lifetime, i.e. the time during which all targets are monitored.

In this section, we describe existing works and details of our problem. In section 2, we reformulate the problem so as to solve it with the methods proposed in sections 3 (allocation) and 4 (scheduling). Finally, the proposed contribution is discussed before concluding the paper.

1.1 Related work

Many protocols with energy-conservation in mind have been proposed [5, 7]. Until 2012, a few methods relying on optimization techniques were reported [5]. Rossi et al. [8] proposed a column generation-based algorithm boosted by a genetic algorithm to solve the *maximum network lifetime under bandwidth constraints* (MNLB). However, this method is only suitable for missions involving static targets. To minimize the communication costs due to data reporting, Lin and Lee [2] proposed an algorithm applied to bi-directional moving objects. The authors formulate the problem as a 0-1 integer programming problem and apply a Lagrangean relaxation-based heuristic to solve it. Naderan et al. [6] solved the problem of multiple target coverage by determining the sensing range of each sensor using primal and dual-based algorithms. In [4], a continuous linear programming model which handles both monitoring and reporting is proposed. The model maximizes network lifetime but assumes that targets are static.

1.2 Problem description

In a region, m sensors are randomly deployed to track n mobile targets as points and to report sensing data to a *base station*. A sensor $i \in \{1, \dots, m\}$ is static and has its own sensing range R_i^S , communication range R_i^C and its own initial battery capacity E_i . Each sensor can be in *active* state, during which it can cover targets and consumes e_i^S units of energy per target and per unit of time, or in *sleep* state, in which the energy consumption is zero. Transmitting data to other sensors costs e_i^T and receiving data e_i^R per unit of data. For each target $j \in \{1, \dots, n\}$, we know the position $\mathcal{T}_j(t)$ at each instant t . We require that each target is covered by at least one sensor at any time.

Our method is composed of three steps : discretization, allocation and scheduling. In the first step, we reformulate the problem in terms of sets by splitting the time horizon into time windows and grouping candidate sensors for watching targets in a sequence of sets. Then from this reformulation we deduce two linear programming models to allocate monitoring time and reporting data amount to sensors. Finally we schedule the sensing tasks and the routing plan by solving a sequence of matching problems.

2 Discretization

To solve our problem, it is convenient to divide the total time into time windows for which we can deduce a static subset of candidate sensors covering the moving targets.

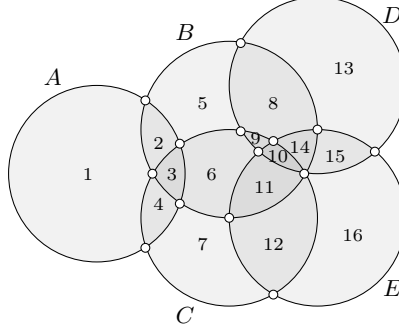


Fig. 1. A planar graph example with $m = 5$ sensors, resulting in 16 faces

In this paper, it is assumed that sensing area of every sensor $i \in \{1, \dots, n\}$ is a disk of radius R_i^S . The monitored area can be seen as a *planar graph* [1, 9] (Figure 1). Vertices are the points of intersections of boundaries of all sensor's disks. Edges connect vertices along the boundaries. The surfaces bounded by edges are called *faces*. In fact, all points inside a face are covered by the same set of sensors.

A circle can intersect at most two times any other circle. Suppose that every circle intersects exactly two times each other, then the number of vertices $|V|$ is at most $m(m-1)$ and the number of edges $|E|$ at most $2m(m-1)$. Thus using the Euler formula $|V| - |E| + |F| = 2$ with $|F|$ the number of faces, $|F|$ is at most $m(m-1) + 2$ (including the outer, infinitely large face) [1].

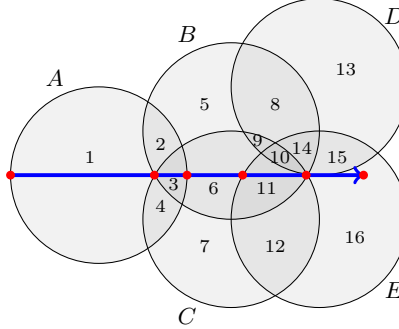


Fig. 2. Temporal discretization in 5 time windows (6 ticks)

We perform a temporal discretization by computing the intersections between each target's trajectory $\mathcal{T}_j(t)$ and the boundaries of the faces (Figure 2). A convenient way to model the trajectories is to use piecewise linear curves, as it can be used to model any trajectory. Computing the intersections in this case is

also equivalent to solve quadratic equations. This computation results to a set of *ticks*, i.e. the values of t for which a target goes from one face to another. Once all targets have been processed, we use all the ticks to split the time, producing p intervals called *time windows*. To each time window $k \in \{1, \dots, p\}$ and each target $j \in \{1, \dots, n\}$ is associated a set $S^k(j)$ of candidate sensors. $S^k(j)$ defines the set of sensors that are able to cover the target j during time window k . To cover the target j during the time window k , at least one sensor has to be activated from $S^k(j)$. Each time window k (defined by $[t_k, t_{k+1}]$) has a duration $\Delta^k = t_{k+1} - t_k$.

3 Allocation

In this section, we propose two models to solve the allocation problem. Precisely, the purpose of these models is to decide how long each sensor monitors which target, and the amount of data sent or received by each sensor. After processing the discretization, we can provide the following data:

- I Set of sensors $\{1, \dots, m\}$.
- J Set of targets $\{1, \dots, n\}$.
- K Set of time windows $\{1, \dots, p\}$.
- $S^k(j)$ Set of sensors covering target j during time window k .
- $T^k(i)$ Set of targets covered by sensor i during time window k .
- $N^T(i)$ Set of sensors that are able to receive data from sensor i (inc. BS).
- $N^R(i)$ Set of sensors that are able to send data to sensor i .
- e_i^S Amount of energy spent by sensor i for sensing task per one unit of time.
- e_i^T Amount of energy spent by sensor i for transmitting one unit of data.
- e_i^R Amount of energy spent by sensor i for receiving one unit of data.
- E_i Battery capacity of sensor i .
- β Amount of data produced per unit of time for sensing.
- Δ^k Duration of time window k .

It can be observed that if there exists a couple (k, j) such that $S^k(j) = \emptyset$, then target j will never be covered during time window k .

We present the two essential sets of decision variables for our models.

- $d_{ij}^k \geq 0$ is the amount of time during which sensor i monitors target j during time window k .
- $f_{ii'}^k \geq 0$ is the amount of data transmitted by sensor i to sensor i' during time window k .

3.1 Energy consumption minimization

The following model assumes that sensors have enough energy to monitor all the targets during the whole time horizon, otherwise it becomes infeasible.

$$\min E = \sum_{i \in I} \sum_{k \in K} \left(\sum_{j \in T^k(i)} e_i^S d_{ij}^k + \sum_{i' \in N^T(i)} e_i^T f_{ii'}^k + \sum_{i' \in N^R(i)} e_i^R f_{i'i}^k \right) \quad (1)$$

$$\text{s.t. } \sum_{k \in K} \left(\sum_{j \in T^k(i)} e_i^S d_{ij}^k + \sum_{i' \in N^T(i)} e_i^T f_{ii'}^k + \sum_{i' \in N^R(i)} e_i^R f_{i'i}^k \right) \leq E_i, \forall i \in I \quad (2)$$

$$\beta \sum_{j \in T^k(i)} d_{ij}^k + \sum_{i' \in N^R(i)} f_{i'i}^k = \sum_{i' \in N^T(i)} f_{ii'}^k, \forall k \in K, i \in I \quad (3)$$

$$\sum_{i \in S^k(j)} d_{ij}^k = \Delta^k, \quad \forall k \in K, j \in J \quad (4)$$

$$d_{ij}^k \geq 0, \quad \forall k \in K, i \in I, j \in T^k(i) \quad (5)$$

$$f_{ii'}^k \geq 0, \quad \forall k \in K, i \in I, i' \in N^T(i) \quad (6)$$

We want to minimize the total energy spent by all the sensors (1). Sensors can consume energy for three distinct tasks: sensing, transmitting and receiving data. Constraint (2) ensures that energy spent by each sensor does not exceed its battery capacity. Sensing task produces input data that needs to be transmitted to the base station. Constraint (3) enforces the connectivity and the data transmission to the base station. This constraint is a data flow conservation constraint, i.e. it implies that the amount of sensed and received data is equal to the amount of transmitted data. Constraint (4) ensures that each target is covered by at least one sensor for every time window.

3.2 Network lifetime maximization

When the network has not enough energy to track all the targets during the time horizon, we would like to maximize the network lifetime. We introduce two additional variables y^k and δ^k .

- $y^k \in \{0, 1\}$ is equal to 1 if all targets are monitored during the whole time window k .
- $\delta^k \geq 0$ is a helper fractional variable in order to take into account the tracking duration in the last (incomplete) time window for expressing the network lifetime.

$$\max L = \sum_{k \in K} \Delta^k (y^k + \delta^k) \quad (7)$$

$$\text{s.t. (2) – (3)}$$

$$\sum_{i \in S^k(j)} d_{ij}^k = \Delta^k (y^k + \delta^k), \quad \forall k \in K, j \in J \quad (8)$$

$$\delta^k \leq y^{k-1} - y^k, \quad \forall k \in K (y^0 = 1) \quad (9)$$

$$y^{k-1} \geq y^k, \quad \forall k \in K \setminus \{1\} \quad (10)$$

$$y^k \in \{0, 1\}, \quad \forall k \in K \quad (11)$$

$$\delta^k \geq 0, \quad \forall k \in K \quad (12)$$

$$d_{ij}^k \geq 0, \quad \forall k \in K, i \in I, j \in T^k(i) \quad (13)$$

$$f_{ii'}^k \geq 0, \quad \forall k \in K, i \in I, i' \in N^T(i) \quad (14)$$

The problem objective is to maximize the time during which all the targets are covered by at least one sensor (7). We consider that the network lifetime is defined by the moment where some target is no longer covered by a sensor. Constraint (8) links the d_{ij}^k variables to the y^k and δ^k variables to ensure that all targets are covered during all the network lifetime L . Only one of the δ^k variables is allowed to be strictly positive (9), in particular in the last time window of the tracking lifetime. Constraint (10) enforces tracking continuity by setting the first consecutive y^k variables to 1 and all the following ones to zero.

One of the advantages of our models is their linearity which makes them suitable for solving in LP/MIP solvers to get an optimal solution. An important remark is that our models don't take into account the case when a sensor watching several targets consumes the same amount of energy as watching one target. The two models complement one another, i.e. the second model can be used as a fallback of the first in case the latter is infeasible. In the case that sensors have enough energy to watch all the targets at any time, the second model becomes irrelevant because it would give the time horizon as an optimal objective value, without considering the energy consumption.

4 Scheduling

4.1 Sensing tasks

The previous step helps us to know which amount of time each sensor should watch targets, but does not say when the sensing tasks should start and stop.

The values of d_{ij}^k obtained in the optimal solution of the LPs, can be casted into a series of matrices called *workload matrices*. The basic idea to determine a schedule is to decompose each of these matrices as a sequence of q *schedule matrices* [3]:

$$D^k = \begin{bmatrix} d_{1,1}^k & d_{1,2}^k & \cdots & d_{1,n}^k \\ d_{2,1}^k & d_{2,2}^k & \cdots & d_{2,n}^k \\ \vdots & \vdots & \ddots & \vdots \\ d_{m,1}^k & d_{m,2}^k & \cdots & d_{m,n}^k \end{bmatrix} = \begin{bmatrix} 0c_1 0 \dots 0 \\ c_1 0 0 \dots 0 \\ 0 0 0 \dots c_1 \\ \vdots \\ 0 0 c_1 \dots 0 \end{bmatrix} + \begin{bmatrix} 0 0 0 \dots 0 \\ 0 0 0 \dots c_2 \\ 0 c_2 0 \dots 0 \\ \vdots \\ c_2 0 c_2 \dots 0 \end{bmatrix} + \cdots + \begin{bmatrix} c_q 0 0 \dots 0 \\ 0 0 c_q \dots c_q \\ 0 0 0 \dots 0 \\ \vdots \\ 0 c_q 0 \dots 0 \end{bmatrix}$$

$$= P_1 + P_2 + \cdots + P_q$$

Each entry in *schedule matrix* P_i is either c_i or 0. Each column has exactly one c_i element (i.e. a target is watched by exactly one sensor).

To do the decomposition, we express the matrix D^k as a bipartite graph connecting sensors i (in one side) to targets j (in the other side). For each non-zero d_{ij}^k , there is an edge connecting i to j of weight d_{ij}^k .

The problem of finding a schedule matrix is equivalent to finding a n -matching in the bipartite graph. As long as the right hand side value is strictly positive, constraints (4) and (8) guarantee that such a matching exists, i.e. that each target is connected to at least one sensor in the bipartite graph. We find a matching by selecting one adjacent edge per target j , denoted by $\text{sel}(j)$.

Let c be the minimum weight over all selected edges, then we subtract c to the weight of these selected edges. Let \mathfrak{L} be the right hand side value of the equations (4) or (8). After this subtraction, the following equations:

$$\sum_{i \in S^k(j)} d_{ij}^k = \mathfrak{L}, \forall j \in J \quad (15)$$

become:

$$\sum_{i \in S^k(j) \setminus \{\text{sel}(j)\}} d_{ij}^k + (d_{\text{sel}(j),j}^k - c) = \mathfrak{L} - c \implies \sum_{i \in S^k(j)} d_{ij}^{k'} = \mathfrak{L}', \forall j \in J \quad (16)$$

Thus the underlying equations keep their original structure with the same right hand side value for all $j \in J$. The edge with the minimal weight c is then removed from the bipartite graph. Matchings can be found until the right hand side \mathfrak{L} is zero. The sequence of matrices can be scheduled using any ordering during the time window.

4.2 Data routing

As a result of our LPs, we obtain a sequence of flow matrices $F^k = (f_{ii'}^k)_{m \times (m+1)}$ that express a sequence of trees where the root is the base station. Each active sensor needs to forward its sensed data to the base station. The method proposed in [3] is to forward the data through non-zero edges (i, i') (i.e. such that $f_{ii'}^k > 0$) to the the base station. A sensor i sends its outgoing data to its first available neighbor i' until the edge is saturated (amount of data $f_{ii'}^k$ reached), then switches to another neighbor i'' until the value $f_{iii''}^k$ is met, etc. There is no specific ordering to follow to build an optimal routing plan.

5 Conclusion

We provide reformulations as a linear model to solve multiple mobile target tracking problems in WSNs. The basic problems considered are energy consumption minimization and network lifetime maximization. Our formulations can be used to evaluate the results of scheduling-based protocols using the optimal solution. Further research may focus on the following perspectives : decentralize the algorithm (swarm intelligence), deal with uncertainty (stochastic optimization) or improve the centralized algorithm. This work is sponsored by the *Direction Générale de l'Armement* (General Directorate for Armement of France).

Bibliography

- [1] Piotr Berman, Gruia Calinescu, Chintan Shah, and Alexander Zelikovsky. Power efficient monitoring management in sensor networks. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 4, pages 2329–2334. IEEE, 2004.
- [2] Frank Yeong-Sung Lin and Cheng-Ta Lee. An efficient lagrangean relaxation-based object tracking algorithm in wireless sensor networks. *Sensors*, 10(9): 8101–8118, 2010.
- [3] Hai Liu, Xiaowen Chu, Yiu-Wing Leung, Xiaohua Jia, and Peng-Jun Wan. Maximizing lifetime of sensor-target surveillance in wireless sensor networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6. IEEE, 2009.
- [4] Hai Liu, Xiaowen Chu, Yiu-Wing Leung, Xiaohua Jia, and Peng-Jun Wan. General maximal lifetime sensor-target surveillance problem and its solution. *Parallel and Distributed Systems, IEEE Transactions on*, 22(10):1757–1765, 2011.
- [5] Marjan Naderan, Mehdi Dehghan, Hossein Pedram, and Vesal Hakami. Survey of mobile object tracking protocols in wireless sensor networks: a network-centric perspective. *International Journal of Ad Hoc and Ubiquitous Computing*, 11(1):34–63, 2012.
- [6] Marjan Naderan, Mehdi Dehghan, and Hossein Pedram. Primal and dual-based algorithms for sensing range adjustment in wsns. *The Journal of Supercomputing*, pages 1–21, 2013.
- [7] K Ramya, K Praveen Kumar, and V Srinivas Rao. A survey on target tracking techniques in wireless sensor networks. *International Journal of Computer Science and Engineering Survey*, 3(4), 2012.
- [8] André Rossi, Alok Singh, and Marc Sevaux. Column generation algorithm for sensor coverage scheduling under bandwidth constraints. *Networks*, 60(3):141–154, 2012.
- [9] Sasha Slijepcevic and Miodrag Potkonjak. Power efficient organization of wireless sensor networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 2, pages 472–476. IEEE, 2001.

Robustness Analysis of Multi-Agent Patrolling Strategies using Reinforcement Learning

Fabrice Lauri and Abderrafaa Koukam

IRTES-SeT

Rue Thiery-Mieg, 90010 Belfort

{fabrice.lauri,abder.koukam}@utbm.fr

Abstract. Patrolling an environment involves a team of agents whose goal usually consists in continuously visiting the most relevant areas as fast as possible. In this paper, we follow up on the work by Santana *et al.* who formulated this problem in terms of a reinforcement learning problem, where agents individually learn an MDP using Q-Learning to patrol their environment. We propose another definition of the state space and of the reward function associated with the MDP of an agent. Experimental evaluation shows that our approach substantially improves the previous RL method in some situations (graph topology and number of agents). Moreover, it is observed that such an RL approach is able to cope efficiently with most of the situations caused by the removal of agents during a patrolling simulation.

1 Introduction

The multi-agent patrolling problem has been rigourously addressed only recently [1–7]. In these works, many patrolling strategies have been devised and experimentally validated using common evaluation criteria [1]. They are based on different approaches, ranging from heuristic laws enabling agents to better choose the next node to visit [1], negotiation mechanisms [2], reinforcement learning techniques [3], techniques based on graph theory [4] to techniques based on ACO [5–7]. Most of these solutions yield good empirical results on different graphs constituted from less than fifty nodes and one hundred edges. Nevertheless, none of these solutions have been evaluated in terms of robustness. In some applications though, one might also want to know how the performances of one of these solutions are influenced by an *online* change in the size of the population of the individuals (or agents) involved in a patrol.

In this paper, we first propose an improvement of the learning agents' architecture presented in [3], by characterizing more precisely the MDP employed by an agent. Moreover, we experimentally show that a reinforcement learning based approach can be efficiently applied to the multi-agent patrolling problem for dealing with both an increase in the graph complexity and an online addition or removal of agents. We chose the reinforcement learning framework for studying the robustness of the patrolling problem for two main reasons. On the one hand, a machine learning approach can theoretically cope with any graph

topology and any agents' set, so that a larger range of situations can be considered. On the other hand, we assume that all the agents are located at the same node at the initial time. Under this condition, the patrolling task starts with a preliminary phase where agents spread out in the graph. This step cannot be handled by the most efficient techniques based on Single Cycle [4, 2], which limits agents to be located at different nodes.

The remainder of this paper is organized as follows. Section 2 describes the commonly used framework of a patrolling problem and gives an overview of the related works. Section 3 reviews the fundamental concepts of a reinforcement learner. Section 4 proposes a specification of the patrolling problem in terms of a reinforcement learning problem. Experimental results are shown in section 5. Finally, concluding remarks and future research issues are given in section 6.

2 Problem Definition

The patrolling problem is usually specified formally as follows [1, 4, 3]. The environment to patrol is reduced to a graph $G = (V, E)$, V representing the strategically relevant areas and E the safe ways of movement or communication between them. A cost c_{ij} , associated with each edge (i, j) , measures the time required to go from node i to node j . Let be r agents bound to visit at regular intervals the areas defined in the graph G . Each agent is located at one of the nodes of V at the initial time. Solving the patrolling problem consists of elaborating a multi-agent graph coverage strategy π . Such a strategy must optimize a given quality criterion. $\pi = \{\pi_1 \cdots \pi_r\}$ is made up of the r individual strategies π_i of each agent i . An individual strategy π_i is defined such that $\pi_i : \mathbb{N} \rightarrow V$, $\pi_i(j)$ denoting the j -th node visited by the agent i .

Intuitively, a relevant patrolling strategy is one that minimizes, for each node, the time span between two visits to the same node. Several criteria have been devised in [1] in order to evaluate the quality of a multi-agent patrolling strategy after T time steps (or *cycles*) of simulation. All of them are based on the notion of *instantaneous node idleness* (INI). The INI $I_t(i)$ of a node i at time t is the number of time steps this node remained unvisited. By convention, at the initial instant, $I_0(i) = 0$, $\forall i = 1, 2, \dots, |V|$. At a given instant t , GI_t is the *instantaneous average graph idleness* (IGI). Similarly the *instantaneous worst graph idleness* WI_t is the highest INI encountered since t time steps of simulation. A multi-agent patrolling strategy π can be evaluated after T cycles of simulation using either the *average idleness* criterion AI_π or the *worst idleness* WI_π . The average idleness denotes the mean of the IGI over the T simulation cycles, whereas the *worst idleness* is the highest INI observed during the T -time steps of the simulation. As emphasized by [4], the optimal strategy π is the one that minimizes the worst idleness, as $WI_\pi \geq AI_\pi$ for any strategy π .

3 Reinforcement Learning Framework

Reinforcement learning typically deals with problems where one or several agents interact with their environment to learn to perform a task. At each time step, an

agent is able to (1) perceive the state of its environment, (2) carry out an action which modifies the environment state and (3) obtain an immediate reward depending on the action it just performed. After several thousands of trials, such an agent learns a policy π , which tells him what to do in every situation [8]. A reinforcement learning problem involving one agent is usually defined in terms of a *Markov Decision Process* (MDP). Several extensions of an MDP, such as MMDP [9] or DEC-MDP [10] have been proposed to deal with the problem of coordination in multi-agent systems, but these solutions are intractable when the number of agents is high. Indeed, they use joint actions whose number exponentially increases with the number of individual actions and agents involved: if there are n agents, each of which can perform a actions, then the size of the joint action space is a^n . To alleviate this problem, many approaches [11–13, 3] consider RL agents as *independent learners*. Independent learners ignore the actions and rewards of the other agents, and learn their policy using their own MDP. Although these approaches are no longer assumed to find a globally optimal solution, they still yield satisfactory results in practice. For this reason, as the patrolling problem may involve a lot of agents, we will focus in this paper on the case where agents employ an MDP learned with Q-Learning to perform its task.

4 Learning to Patrol using Reinforcements

One of the most difficult tasks when designing a patrolling agent’s MDP is the definition of its state space. As each of our agents uses partial information to find a globally optimal solution to the patrolling problem, the more features are incorporated in a state, the more precise the solution can be. On the other hand, it is well known that the size of the state space grows exponentially with the number of features. Defining the state space of the MDPs is thus a trade-off between their computational complexity and the global solution approximation they yield. In [3], the learning agents’ architecture which obtained the best results was *Gray-Box Learner Agent* (GBLA) when using the idleness of the next reached node as the immediate reward. This architecture incorporates into an agent’s MDP some information characterizing its environment vicinity and enables each of them to communicate its intention about its next action. As this architecture constituted the first attempt to formulate the patrolling problem in a reinforcement learning framework, it is unfortunately not perfect. In the next sections, we will discuss the drawbacks of this architecture and see how its definition can be refined.

4.1 Identifying the dark side of MDP

Considering that d stands for the graph degree and $|\mathcal{V}|$ is the number of nodes of the graph, the state space S in GBLA was made up of the following components : (1) the node where the agent is ($|\mathcal{V}|$ possible values) (2) the edge from which it came (d possible values) (3) the neighbor node which has the highest (worst) idleness (d possible values) (4) the neighbor node which has the lowest idleness

(d possible values) and (5) the list of the adjacent nodes which are intended to be visited by other agents (2^d possible values). The cardinality of the action set was equal to the graph degree d , each action enabling an agent to reach an adjacent node. As emphasized previously, the size of the state space grows exponentially with the number of features : with this MDP definition, the total number of states $|S| = |\mathcal{V}| \times d^3 \times 2^d$ and the total number of actions $|A| = d$. Learning several MDPs (one for each patrolling agent) with Q-Learning can therefore become rapidly intractable when a lot of agents try to patrol in a graph of high degree and a great number of nodes. For instance, with a graph with a degree of 7 and constituted by 50 nodes (the graph called *map A* in the previous works), each MDP needs theoretically to store more than 15 million scalars (used by the Q-table).

4.2 Numbering the valid states using the graph topology

Yet, among the $|\mathcal{V}| \times d^3 \times 2^d$ states, a lot of them will never be visited by an agent. For instance, let us consider the five-node graph $G = (V, E)$, where $V = \{1, 2, 3, 4, 5\}$ and $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{4, 5\}\}$. Here, the MDP of an agent patrolling this graph will be potentially made up of $|S| = 5 \times 3^3 \times 2^3 = 1080$ states. But some states will never be encountered by the agent, such as the ones in which it is in node 3 and came from a node reached when performing action 2 in node 3. In fact, the number of states that will be effectively visited by an agent (the *valid states*) can be computed precisely from the graph topology. Assuming that d_i is the degree of node i , the number of valid states induced by this MDP definition is equal to $|S| = \sum_{i=1}^{|\mathcal{V}|} d_i^3 \times 2^{d_i}$. Using this formula, the size of the state space associated to this graph is reduced to $|S| = 252$, which is about one quarter of the size of the initial state space for this graph topology. Using this simple principle, since the number of states is now reduced to a sum over the total number of graph nodes, graphs with a greater amount of nodes can be dealt with. This reasoning can be pushed further by considering that only d_i actions at node i can be performed. The number of valid indices (s, a) can thus similarly be reduced when a particular scalar must be accessed through the Q-table. For instance, when dealing with map A, no more than 1.6 million scalars would need to be stored, which is here nearly one tenth of the size of the initial state space. In order to take into account only the valid states and thus avoid allocating too much memory, a variant of Q-Learning will be used to learn each MDP. This version of Q-Learning maintains an ordered list of the states that have been visited at least once. When $Q(s, a)$ must be accessed from a state s and an action a (through the Q-table), a search of the corresponding valid state s' is initiated in the *already visited states* list. If state s does not exist in it, it is added at the end. Else, its order number s' in the list is used as the first part of the index for the Q-table.

4.3 Adding more local information in a state representation

We just saw that the topology of the graph to be patrolled can considerably reduce the size of the state space, so that a better characterization of the envi-

ronment vicinity and of the information required to coordinate the agents' action can possibly be incorporated into a state vector. Indeed, it seems to us that the MDP defined in GBLA was incomplete. It is incomplete because the third and fourth features of a state (the neighboring node which has the highest idleness and the neighboring node which has the lowest idleness) do not precisely inform the agent about its environment vicinity if more than two edges are connected to a node. In order to enable an agent to decide which is the best action to execute in a given state using the most relevant information, we redefined the MDP associated to an agent in terms of its state space and reward function. Firstly, we suggest to represent the following features on the state : (1) the node where the agent is ($|\mathcal{V}|$ possible values), (2) the edge from which an agent came from (d_i possible values), (3) an ordered list of the adjacent nodes from node i , sorted according to their idleness ($d_i!$ possible values), (4) the list of the adjacent nodes from node i which are intended to be visited by other agents (2^{d_i} possible values). The number of valid states induced by this state space is equal to $|\mathcal{S}| = \sum_{i=1}^{|\mathcal{V}|} d_i \times d_i! \times 2^{d_i}$. Secondly, the immediate reward given to an agent is equal to zero if the reached node was bound to be visited by other agents, else it is equal to the idleness of the reached node. This new model will be called Extended-GBLA in the remainder of this article.

5 Experimental Results

Multi-agent patrolling strategies were trained on the six different graph topologies commonly used by the community, with populations of 2 to 15 agents.

To obtain strategies as robust as possible, their training were divided into trials. At each trial, graph statistics (the node idlenesses and the average graph idleness) was set to zero, all the agents were placed at the same starting node and they learned to patrol during several iterations. The starting node changed from one trial to the other. Patrolling strategies were trained using either GBLA or Extended-GBLA. Thus, a total of 120 patrolling strategies (10×6 for each RL method) were trained. Preliminary experiments were conducted to determine the learning parameters, such as the number of trials, the number of iterations per trial, the learning rate α , the discount factor γ and the exploration probability ϵ . The agents' MDPs were trained using 1000 trials, 10000 iterations per trial, $\alpha = 0.9$, $\gamma = 0.9$ and $\epsilon = 0.1$. Two classes of experiments were carried out to assess the robustness of our multi-agent patrolling strategies. The first ones were conducted in order to know whether the patrolling strategies trained with GBLA or with Extended-GBLA are still efficient when the node where all agents start to patrol is changed. The second experiments measure the capacity of the patrolling agents to adapt on situations where some agents broke down.

5.1 Comparison of GBLA and Extended-GBLA

Figure 1 presents the average graph idleness obtained after a multi-agent patrolling simulation using strategies trained with GBLA and Extended-GBLA.

Each trained patrolling strategy was evaluated 20 times by changing the starting node of agents and by using 50000 cycles of simulation. Thus, subsequent results represent the average graph idleness over the 20 runs. Confidence intervals indicated on figures were computed using a risk of 5%. One can already see

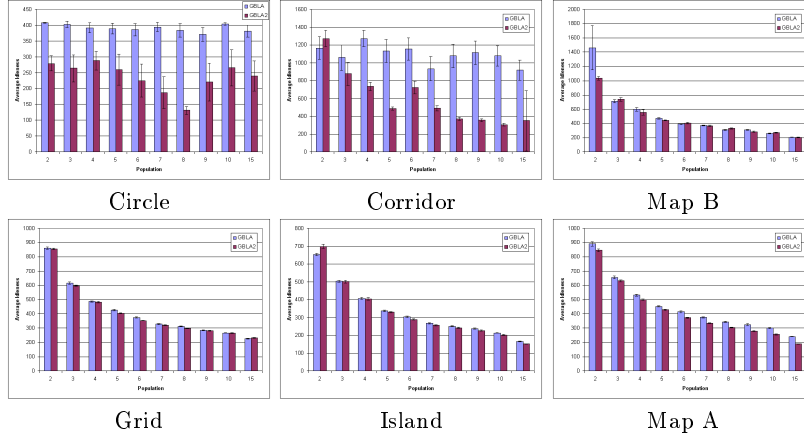


Fig. 1. Comparison results between GBLA and Extended-GBLA

that for the Map A, the Map B, the Grid Map and the Island Map, agents have learned to coordinate their actions, since the average graph idleness decreases when the number of agents increases. Despite their lowest degree, patrolling on the Corridor-shaped graph and on the Circle-shaped graph seems to be more complicated. Both RL methods give equivalent performance for the Map B, the Grid Map and the Island Map. For the other graphs, Extended-GBLA is significantly better than GBLA. By observing the agents' behavior on our simulator, we classified patrolling agents into two different classes. The first class is composed of agents that are responsible of only one region of the graph: they patrol only nodes of that region during the whole simulation. The second class is made up of agents that cross from one region to another one, especially to visit the node which links several regions, thus avoiding to decrease performances. These behaviors were only observed on the four more complex graphs (Map A, Map B, Island and Grid) with both GBLA and Extended-GBLA, and on the two lowest complex graphs (Corridor and Circle) only with Extended-GBLA. With GBLA, all the agents follow the same politics on the latter graphs : they all cross the graphs in the same direction and at the same time. This explains why the average graph idleness does not decrease when the agents' population grows. It is not the case using Extended-GBLA. We explain this phenomenon by considering that with Extended-GBLA, agents are informed of the utility to go to a given node through the reward function: if one agent intends to visit a node, the other agents will not want to visit it as it will give a zero reward. Hence, the definition

of the reward function of Extended-GBLA enables agents to better coordinate their action. From this point of view, we can say that the information added by Extended-GBLA to the state space of MDPs used by agents (that is the ordered list of the adjacent nodes sorted according to their idleness) seems to have less influence than reward functions do on the performance of the patrolling strategies.

5.2 Robustness of Extended-GBLA

Figure 2 shows the influence of the removal of agents on the average graph idleness obtained after a 100000-cycle patrolling simulation using multi-agent strategies trained with Extended-GBLA. These experiments were carried out on the Corridor map and on the Map B with 5 and 10 agents. One such experiment consisted of choosing how many agents will be removed (or will fall into a breakdown) during the simulation, considering that the first agent will be removed after the 10000th cycle and that each subsequent agent will be removed every 10000 cycles. Results show that for both graphs, strategies are no more efficient

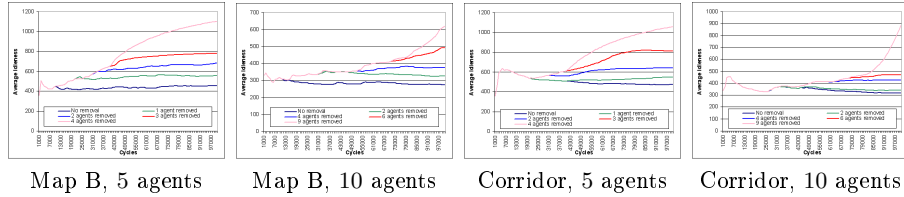


Fig. 2. Robustness results of Extended-GBLA

when only one agent remains to patrol. Indeed, as an agent has only a local representation of its environment when it uses an MDP, it often forgets to visit some nodes for a while, thus decreasing performance. This forgetting behavior can also be observed when more than one agent remains to patrol (for instance on Map B with 10 agents where 6 agents are removed). In this case, it is due to the specialization of some agents that learned to patrol only in a given area of the graph. In the other cases, when a sufficient number of agents are patrolling, one can observe that agents are able to rapidly adapt to the new situations.

6 Concluding Remarks and Future Works

We have proposed in this article a novel definition of MDPs used by agents to learn individually how to patrol in a graph. The RL algorithm Q-Learning was used to give agents the capability to select the best actions to carry out in a given situation in a dynamic environment (that is where agents continuously move). We have experimentally shown that our RL method Extended-GBLA significantly outperforms in several graph topologies the approach proposed by

Santana *et al.* [3]. We believe this improvement is mainly due to the redefinition of the reward function, which allows agents to better coordinate their actions. Moreover, results evaluating the robustness of Extended-GBLA reveal that the patrolling strategies trained with this method are still efficient when some agents are removed during a patrolling simulation. However, the trained strategies were unable to cope adequately with situations where only a few agents remain to patrol. To tackle this problem, an adaptation phase seems to be required to allow the remaining patrolling agents to face new situations caused by a removal of a lot of agents. Future research directions include the use of an undiscounted RL method, such as R-Learning [14], to better characterize this problem by optimizing the average reward.

References

1. Machado, A., Ramalho, G., *al*: Multi-Agent Patrolling : an Empirical Analysis of Alternatives Architectures. In: 3rd International Workshop on Multi-Agent Based Simulation. (2002) 155–170
2. Almeida, A., Ramalho, G., *al*: Recent Advances on Multi-Agent Patrolling. In: 17th Brazilian Symposium on Artificial Intelligence. (2004) 474–483
3. Santana, H., Ramalho, G., *al*: Multi-Agent Patrolling with Reinforcement Learning. In: 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems. (2004) 1122–1129
4. Chevaleyre, Y.: Theoretical Analysis of the Multi-Agent Patrolling Problem. In: International Joint Conference on Intelligent Agent Technology. (2004) 302–308
5. Lauri, F., Charpillat, F.: Ant Colony Optimization applied to the Multi-Agent Patrolling Problem. In: IEEE Swarm Intelligence Symposium. (2006)
6. Lauri, F., Koukam, A.: A Two-Step Evolutionary and ACO Approach for Solving the Multi-Agent Patrolling Problem. In: IEEE World Congress on Computational Intelligence. (2008)
7. Lauri, F., Koukam, A.: Hybrid ACO/EA Algorithms applied to the Multi-Agent Patrolling Problem. In: IEEE World Congress on Computational Intelligence. (2014)
8. Sutton, R., Barto, A. In: Reinforcement Learning : An Introduction. Cambridge, MA (1998)
9. Boutilier, C.: Sequential Optimality and Coordination in Multi-Agent Systems. In: 16th International Joint Conference on Artificial Intelligence. (1999) 478–485
10. Bernstein, D., Zilberstein, S., Immerman, N.: The Complexity of Decentralized Control of Markov Decision Processes. In: 16th Conference on Uncertainty in Artificial Intelligence. (2000) 32–37
11. Sen, S., Sekaran, M., Hale, J.: Learning to coordinate without sharing information. In: 12th National Conference on Artificial Intelligence. (1994) 426–431
12. Schneider, J., Wong, W., Moore, A., Riedmiller, M.: Distributed value functions. In: 16th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA (1999) 371–378
13. Wolpert, D., Wheeler, K., Tumer, K.: General Principles of Learning-based Multi-Agent Systems. In: 3rd International Conference on Autonomous Agents (Agents'99). (1999) 77–83
14. Schwartz, A.: A reinforcement learning method for maximizing undiscounted rewards. In: 10th International Conference on Machine Learning. (1993) 298–305

A new ACO for solving a hybrid flexible flowshop with sequence-dependent setup times

Aymen Sioud, Caroline Gagné, and Marc Gravel

Département d'informatique et de mathématique
Université du Québec à Chicoutimi.
555 Boulevard Université, Chicoutimi, Canada
{asioud, c3gagne, mgravel}@uqac.ca

Abstract. *In this paper, we propose a new ACO to solve a hybrid flexible flowshop problem with sequence-dependent setup times under the objective of minimizing the makespan. The proposed ACO uses concept from multi-objective evolutionary algorithms and look-ahead information to enhance solutions quality. We compare the performance of the introduced ACO on benchmark from the literature. The experiments show that the ACO is very competitive and enhances solutions of the known reference sets.*

Keywords: scheduling, ant colony optimization, sequence-dependent setup, flowshop, makespan, hybridization

1 Introduction

In many industries including pharmaceutical, metallurgical production, electronics, ceramics and automotive manufacturing, there are frequently setup times on equipment between two different activities. These setup times can be or not sequence dependent. Dudek et al. [5] reported that 70% of industrial activities include dependent setup times. More recently, Conner [2] point out in 250 industrial projects that 50% of these projects contains setup dependent times and when these setup times are applicable, 92% of the order deadline could be met. Production of good schedules often relies on management of these setup times [1].

Furthermore, the real production systems rarely employ a single machine. Therefore, in many times, the regular flowshop problem is extended with a set of usually identical parallel machines at each stage, i.e., instead of having a series of machines, we have a series of stages. The goal here is to increase the capacity and the outflow of the production system and to reduce the impact of bottleneck stages on the overall shop efficiency. It is also frequent in practice to have optional treatments for products, like polishing or additional decorations in ceramic manufacturing as an example [18]. In this latter case some jobs will skip some stages. Thereby, we obtain the hybrid flexible flowshop.

This present paper considers the sequence dependent setup times hybrid flexible flowshop problem (SDST/HFFS) with the objective of minimizing the makespan. In accordance with the notation for hybrid flowshops of Vignier at

al. [23] who extend the well-known three fields notation $\alpha/\beta/\gamma$ of Graham et al. [7], this NP-hard problem is noted as $((PM)^{(i)})_{i=1}^m/F_j, s_{ijk}/C_{max}$.

The $((PM)^{(i)})_{i=1}^m/F_j, s_{ijk}/C_{max}$ problem may be defined as a set of N jobs, $N=\{1, \dots, n\}$ available for processing at time zero on a set of M stages, $M=\{1, \dots, m\}$. At every stage i , $i \in M$, we have a set of M_i , $M_i=\{1, \dots, m_i\}$ of identical parallel machines. Every machine at each stage can process all the jobs. Each job has to be processed in exactly one out of the M_i identical parallel machines at stage i . However, some jobs will skip some stages. F_j denotes the set of stages that the job j , $j \in N$ has to visit. Furthermore, only stage skipping is allowed, so is not possible that a given job visits stages $\{1, 2, 3\}$ and another one visits stages $\{3, 2, 1\}$. p_{ij} denotes the processing time of job j at stage i . Finally, s_{ijk} denotes the setup time between jobs j and k , $k \in N$ at stage i . The optimization criterion is the minimization of the maximum completion time or makespan calculated as $C_{max} = \max_{j \in N} \{C_j\}$.

In this work, to solve the SDST/HFFS problem, we introduce a new ant colony optimization algorithm (ACO) which incorporates a transition rule that having feature using look-ahead information based on heuristic and past information based on archive concept such as the multiobjective evolutionary computation. The body of this paper is organized into five sections. Section 2 provides a brief literature review of the SDST/HFFS problem. Section 3 describes the proposed ACO algorithm. The computational testing and discussion are presented in Section 4. Finally, we conclude with some remarks and future research directions.

2 Literature review

Only a few papers addressed SDST/HFFS problem. However, to our knowledge, there seemingly exist only six works proposing heuristics and/or metaheuristics for the problem of this paper. Kurz and Askin [12] introduce dispatching rules based on greedy methods, insertion heuristic and an adaption of Johnson's rule. Afterward the same authors formulate an integer programming (IP) model and develop random keys genetic algorithm (RKGA) [13]. The results show that the IP model have difficulties to solve the SDST/HFFS problem and that the RKGA outperforms the dispatching rules of Kurz and Askin [12] and other heuristics. All the algorithms are tested on generated problem data. Zandieh et al. [25] propose an immune algorithm (IA) outperforms the RKGA of Kurz and Askin [13]. The authors use a real representation for individuals and the OX crossover as crossover operator. Mirsanei et al. [14] propose a simulated annealing (SA) using pair wise and inverse interchange as moving operators. The authors also used the SPT Cyclic Heuristic of Kurz and Askin [13]. They show that the SA outperforms the RKGA of Kurz and Askin [13] and the IA of Zandieh et al. [25]. Naderi et al. [15] propose a dynamic dispatching rule heuristic and an iterated local search (ILS). The author propose 960 test instances and compare their approaches to the dispatching rules and the RKGA of Kurz and Askin [12, 13], the IA of Zandieh et al. [25] and the GA of Ruiz et Maroto [18] which

is used for a different problem. The results show that their ILS with different encoding scheme gives better results than all the other algorithms. Gasquet et al. [6] propose an agent-based genetic algorithm using the S2BOX crossover of Ruiz et Maroto [18]. In this latter study the authors focus on introducing the agent-solution scheme to solve the SDST/HFFS problem.

3 The ACO algorithm

The ant colony optimization (ACO) is a population based metaheuristic designed to solve combinatorial optimization problems introduced by Dorigo [3] and inspired by studies of the behavior of the ants. Dorigo and Gambardella [4] propose notably improvements to the original ACO version. The improvements include a modified transition rule called the pseudo-random-proportional rule, global and local trail updating rules, use of restricted candidates list and the use of local improvement rules. To solve the SDST/HFFS problem with the objectives of minimizing the makespan and after the pheromone initialization, the ACO algorithm evolved in five steps in the main loop where a ants construct a sequence of N jobs : (i) an initial job is set pseudo-randomly; (ii) each ant builds a job sequence using the pseudo random proportional transition rule in Equations (1) and (2) ; (iii) a local pheromone update is performed ; (iv) a local improvement heuristic is applied ; and (v) a global pheromone update is applied. This main loop is executed for t_{max} cycles as shown in Figure 1.

From an existing partial job sequence each ant builds a sequence using the pseudo random proportional transition rule in Equations (1) and (2). In Equation (1), q is a random number and q_0 is a parameter; both are between 0 and 1. The parameter q_0 determines the relative importance of the existing information exploitation and the new solutions search space exploration. Indeed, Equation (1) states that the next job will be chosen by a greedy rule when $q \leq q_0$ or by the probabilistic rule of Equation (2) when $q > q_0$. Equation (2) describes the biased exploration rule p_{ij} also adapted to the $((PM)^{(i)})_{i=1}^m / F_j, s_{ijk} / C_{max}$ problem when inserting job j after job i .

In these equations, the elements $\tau_{ij}(t)$ and η_{ij} represent the pheromone trail and the visibility, respectively. Concerning the visibility, η_{ij} represents the inverse of the largest completion time among all the jobs in the list of the unselected jobs. Obviously, the completion time includes the setup times between the last scheduled job and the next one. The element $SUCC_{ij}(A_t)$ represents the past information which is introduced by a matrix built from an archive that stores the best solutions throughout the evolution process as in some cases in multi-objective evolutionary algorithms using the Pareto-optimal concept. This concept plays a role of a long-term memory. From the archive we build a matrix which computes the times that a job j follows a job i in the archive solutions. Finally, the element H_{ij} represents the look-ahead information which uses an heuristic that anticipates the choices in the transition rule. This heuristic is based on an upper bound of the makespan and uses the average values of processing time $\overline{p_{ij}}$ and the normalized setup times $\overline{s_{ijk}}$ for the unscheduled jobs.

```

/* STEP 0 : Pheromone Initialization */
for all job pair (i,j) do
     $\tau_{ij}(0) = \tau_0$ 
end for
/* Main Loop */
for  $t = 1 \rightarrow t_{max}$  do
    /* STEP 1 : Set initial job */
    for  $k = 1 \rightarrow m$  do
        Set the initial job for the ant k
    end for
    /* STEP 2 : Build a sequence */
    for  $i = 2 \rightarrow n$  do
        for  $k = 1 \rightarrow m$  do
            Choose the next job using the Equations 1 and 2
        /* STEP 3 : Local pheromone update */
        for all chosen job pair (i, j) do
             $\tau_{ij}(t) = p_t * \tau_{ij}(t) + (1 - p_t) * \Delta\tau_{ij}(t)$  where  $\Delta\tau_{ij} = \tau_0$ 
        end for
    end for
    /* STEP 4 : Local improvement */
    for  $k = 1 \rightarrow m$  do
        Apply local improvement method or-opt heuristic
    end for
    /* STEP 5 : Global pheromone update */
    for all adjacent job pair (i, j)  $\in$  the best sequence  $Q^*$  do
         $\tau_{ij}(t) = p_t * \tau_{ij}(t) + (1 - p_t) * \Delta\tau_{ij}(t)$  where  $\Delta\tau_{ij} = 1/L^*$ 
    end for
end for

```

Fig. 1. The ACO algorithm

After computing the makespan of the generated sequence by the ant, we apply a local improvement under probability p_{LI} . We use for that a new constructive heuristic. This heuristic is defined as follows : (i) a crosspoint is chosen randomly (as one point crossover operator); (ii) all the jobs in the job set following this crosspoint, noted as J_u , are substitute by jobs with the average values of processing time $\bar{p}_{i\bar{j}}$ and the normalized setup times $\bar{s}_{i\bar{j}k}$ for the J_u jobs, than the makespan is calculated; and (iii) from the last scheduled job, at each time, we insert a job from J_u minimizing the makespan until fulfill the sequence.

The trail pheromone is initialized to the value $\tau_0 = (N * L_r)^{-1}$ where N is the jobs number and L_r is the makespan value of a randomly generated sequence. The other parameters have been assigned the following values $p = p_t = 0.9$, the ant number $a = 10$ and $q_0 = 0.9$. The parameters α , β , ϕ and δ associated with the four matrices in the transition rule were set to identical values for all the problems. These parameters were adjusted following empirical tests on different instances. The four parameters τ , α , β and δ have been assigned the values

$$j = \begin{cases} \arg \max \left\{ [\tau_{ij}(t)]^\alpha \times [SUCC_{ij}(A_t)]^\beta \times [\eta_{ij}]^\delta \times \left[\frac{1}{H_{ij}}\right]^\phi \right\} & \text{if } q \leq q_0 \\ J & \text{if } q > q_0 \end{cases} \quad (1)$$

where J is chosen according to the probability p_{ij}

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha \times [SUCC_{ij}(A_t)]^\beta \times [\eta_{ij}]^\delta \times \left[\frac{1}{H_{ij}}\right]^\phi}{\sum [\tau_{ij}(t)]^\alpha \times [SUCC_{ij}(A_t)]^\beta \times [\eta_{ij}]^\delta \times \left[\frac{1}{H_{ij}}\right]^\phi} \quad (2)$$

4, 2, 3 and 3, respectively. Finally, the archive size and the local improvement probability p_{LI} have been assigned the values 20 and 0.05, respectively.

4 Computational results and discussion

The benchmark problem set is available from <http://soa.iti.es> and consists of 960 problem tests. The instances are combinations of N and M , where $N = \{20, 50, 80, 120\}$ and $M = \{2, 4, 8\}$. The processing times are generated from a uniform $[1, 99]$ distribution. The setup times are generated according to four distributions $[1, 25]$, $[1, 50]$, $[1, 99]$ and $[1, 125]$. This corresponds to a ratio between setup and processing times of 25%, 50%, 100% and 125%, respectively. Concerning the number of parallel machines at each stage, there is a group with two parallel machines per stage and groups where the number of parallel machines at each stage is sampled from a uniform distribution in the range $[1, 4]$. The probability of skipping a stage for each job is set at 0.10 and 0.40. All the experiments were run on an Intel Core 2.4 GHz processors and 4 GB of main memory.

In their paper, Naderi et al. [15] compare the ILS to several metaheuristics and heuristics as the RKGA of Kurz and Askin [13], the IA of Zandieh et al. [25], the genetic algorithm (GAR) of Ruiz et Maroto [18] and the dispatching rules of Kurz and Askin [12] to cite these methods among other. The authors show that the ILS and the GAR represent the two best methods. It is necessary to address that our experiment environment is different from that of Naderi et al. [15]. Thereby, we use the following website reference [21] and [22] to determine the ratio of performance between the two computers. So, in order to obtain a reliable comparison, all the experiments was done with the stopping criterion set to $n^2 \times m \times 1.5 \times 0.78$ ms elapsed CPU time ([15] used $n^2 \times m \times 1.5 \times 0.78$ ms elapsed CPU time as stopping criterion for all the compared algorithms). To evaluate the different algorithms we use the following performance measure :

$$\% \text{ Increase Over the Best Solution} = \frac{Heu_{sol} - Best_{sol}}{Best_{sol}} \times 100$$

where Heu_{sol} is the best makespan obtained by a given algorithm after 10 executions and $Best_{sol}$ is the best known makespan.

Table 1 compares the results of different approaches. All the presented results show the average deviation to the best known solution for the 960 instances grouped by instance characteristics. The ILS and GAR columns show the results of the iterated local search of Naderi et al. [15] and the genetic algorithm of Ruiz et Maroto [18] noted as GAR, respectively. The ACO column presents the ACO algorithm embedding the transition rule defined in Equation (1) and (2) without the local improvement heuristic. Finally, the ACO_I column presents the ACO algorithm embedding the transition rule defined in Equation (1) and (2) using the new local improvement heuristic. In Table 1, the best average results are bold.

Table 1. Comparison of the ILS [15], the GAR [18], the ACO and the ACO_I

Instance	ILS	GAR	ACO	ACO_I
20×2	1.39	2.61	2.05	1.57
20×4	1.27	2.94	1.95	1.31
20×8	1.49	3.50	1.56	1.53
50×2	1.26	2.51	2.45	1.45
50×4	1.85	4.03	1.97	1.88
50×8	1.75	3.20	1.78	1.71
80×2	2.06	2.15	2.14	2.02
80×4	2.92	4.64	3.05	2.88
80×8	5.46	4.51	4.65	4.33
120×2	3.38	3.42	3.35	3.32
120×4	6.74	5.97	5.94	5.88
120×8	9.25	5.19	6.21	5.02
Average	3.23	3.72	3.10	2.95

The first observation is that the ACO algorithm outperform the GAR algorithm on all the instances except for the 80×8 and 120×8 group instances where the deviation is very small. Also, the ACO algorithm perform better than the ILS on the larger instances. Our hypothesis for these results is that the two elements, the $SUCC_{ij}(A_t)$ element which represents the past information, i.e., is a kind of long term memory and the H_{ij} element which represents the look-ahead (future) information, explore more search space when they have more information. Indeed, the two elements allow to take into account the job positions on the next stages and the job position on the best found sequences.

As can be seen, the ACO_I algorithm which uses the transition rule defined in Equation (1) and (2), and the new local improvement heuristic provides better results among all the other algorithms except for the 20 jobs instance group where the ILS performs little better. Furthermore, the ACO_I algorithm obtain

the best average for seven instance groups (50×8 , 80×2 , 80×4 , 80×8 , 120×2 , 120×4 and 120×8). Comparing to the ILS algorithm, the ACO.I found results very close to the ILS results on the smaller instance groups. Moreover, the ACO.I algorithm have the best average (2.95). So, we can conclude that, in general, the local improvement heuristic is effective.

These results are confirmed by a Wilcoxon signed-rank test [24] experimentations not reported here. We also not report other experimentations to evaluate the impact of each element in the transition rule.

5 Conclusion

In this work, we have introduced an ACO algorithm that integrates archive concept in the transition rule and look-ahead information to solve the hybrid flexible flowshop problem with sequence-dependent setup times minimizing the makespan. Also, we introduce a new constructive heuristic used as improvement method in the ACO. The proposed approach is essentially based on adapting the transition rule to the specifics of the studied problem. The numerical experiments allowed us to demonstrate the efficiency of the proposed approach for this problem, especially for large instance groups.

A perspective of this work is to use this proposed approach for other scheduling problems in particular and other optimization problems in general, specially real-world problems. Also, we will work on the refinement of the look-ahead heuristic to enhance results quality.

References

1. Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y.: A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*. 187(3), 985 – 1032 (2008)
2. Conner, G.: 10 questions. *Manufacturing Engineering Magazine* 93–99 (2009)
3. Dorigo M. : Optimization, Learning and Natural Algorithms. PhD thesis, Politecnico di Milano, Italy (1992)
4. Dorigo M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* (1997)
5. Dudek, R.A., Smith, M.L., Panwalkar, S.S.: Use of a case study in sequencing/scheduling research. *Omega* 2(2), 253–261 (1974)
6. Gomez-Gasquet, P., Andres, C., Lario, F.Z.: An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan. *Expert Systems with Applications* 39(9), 8095 – 8107 (2012)
7. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.G.H.: Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics* 5, 287–326 (1979)
8. Gupta J.N.D.: Two-stage, hybrid flowshop scheduling problem. *The Journal of the Operational Research Society* 39(4), pp. 359–364 (1988)
9. Holland J.H.: *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, (1992)

10. Jabbarizadeh, F., Zandieh, M., Talebi, Z.: Hybrid flexible flowshops with sequence-dependent setup times and achine availability constraints. *Comput. Ind. Eng.* 57(3), 949–957 (2009)
11. Javadian, N., Fattahi, P., Farahmand-Mehr, M., Amiri-Aref M., Kazemi, M.: An immune algorithm for hybrid flow shop scheduling problem with time lags and sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology*. 63, 337–348 (2012)
12. Kurz, M., Askin, R.: Comparing scheduling rules for flexible flow lines. *International Journal of Production Economics*. 85(3), 371 – 388 (2003)
13. Kurz, M., Askin, R.: Scheduling flexible flow lines with sequence-dependent setup times. *European Journal of Operational Research*. 159(1), 66 – 82 (2004)
14. Mirsanei, H.S., Zandieh, M., Moayed, M.J., Khabbazi, M.R. : A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times. *Journal of Intelligent Manufacturing*. 22, 965–978 (2011)
15. Naderi, R., Ruiz, R., Zandieh, M.: Algorithms for a realistic variant of flowshop scheduling. *Comput. Oper. Res.* 37(2), 236–246 (February 2010)
16. B. Naderi, M. Zandieh, and V. Roshanaei. Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. *The International Journal of Advanced Manufacturing Technology*, 41:1186–1198, 2009.
17. R. Ruiz, Serifoglu F. S., and T. Urlings. Modeling realistic hybrid flexible flowshop scheduling problems. *Computers and Operations Research*, 35(4):1151 – 1175, 2008.
18. R. Ruiz and C. Maroto. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169(3):781–800, March 2006.
19. R. Ruiz and T. Stutzle. An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, 187(3):1143 – 1159, 2008.
20. A. Sioud, M. Gravel, and M. Gagné. A hybrid genetic algorithm for the single machine scheduling problem with sequence-dependent setup times. *Computers & OR*, 39(10):2415–2424, 2012.
21. SPEC. Intel core 2 duo p8400 @ONLINE, 2009. <http://www.spec.org/cpu2006/results/res2010q1/cpu2006-20100118-09344.html>.
22. SPEC. Intel core i7-2600 @ONLINE, 2011. <http://www.spec.org/cpu2006/results/res2011q3/cpu2006-20110718-17542.html>.
23. A. Vignier, J.-C. Billaut, and Proust C. Les problemes d’ordonnancement de type flow-shop hybride : etat de l’art. *RAIRO - Operations Research - Recherche Operationnelle*, 33(2):117–183, 1999.
24. F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.
25. M. Zandieh, Fatemi Ghomi S.M.T., and Moattar Hussein S.M. An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation*, 180(1):111 – 127, 2006.

A Fuzzy-Controlled Comprehensive Learning Particle Swarm Optimizer

Mahamed G. H. Omran¹, Maurice Clerc², Ayed Salman³ and Salah Alsharhan¹

¹Department of Computer Science, Gulf University for Science and Technology, Kuwait

{omran.m, alsharhanS}@gust.edu.kw

²Independent Consultant, France

maurice.clerc@writeme.com

³Computer Engineering Department, Kuwait University, Kuwait

ayed.salman@ku.edu.kw

Abstract. An adaptive variant of Comprehensive Learning Particle Swarm Optimizer (CLPSO) is proposed in this paper. The proposed method, called Fuzzy-Controlled CLPSO (FC-CLPSO), uses a fuzzy controller to tune the probability learning, inertia weight and acceleration coefficient of each particle in the swarm. The FC-CLPSO is compared with CLPSO and SPSO2011 on 11 benchmark functions. The results show that FC-CLPSO generally outperformed CLPSO and SPSO2011 on most of the tested functions.

Keywords. Particle Swarm Optimization; Fuzzy Controller; Adaptation; Comprehensive Learning.

1 Introduction

The Particle Swarm Optimizer (PSO) (Eberhart and Kennedy 1995) is a nature-inspired metaheuristic which mimics the behavior of bird flocking and fish schooling. PSO is easy to understand and implement. In addition, it requires no gradient information. PSO has been used to solve many real-world problems (Olsson 2011).

In PSO, each particle represents a candidate solution, which is a point in a D -dimensional space. A particle has a position, a velocity and a cost function. The velocity and position of the j -th dimension of the i -th particle are defined as,

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1r_{1,j}(t)(p_{i,j}(t) - x_{i,j}(t)) + c_2r_{2,j}(t)(g_j(t) - x_{i,j}(t)) \quad (1)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t) \quad (2)$$

where w is the inertia weight, c_1 and c_2 are the acceleration coefficients, $r_{1,j}$ and $r_{2,j}$ are two uniformly distributed random numbers in the interval $[0,1]$ with $j \in \{1,2, \dots, D\}$, $i \in \{1,2, \dots, N\}$ and N is the swarm size. \vec{x}_i is the position of the i -th particle, \vec{p}_i is the personal best position of the i -th particle and \vec{g} is the best position discovered by the swarm.

The basic PSO tends to converge prematurely in multi-modal functions due to poor diversity (Riget and Vesterstrøm 2002). A relatively recent variant of PSO was proposed by Liang *et al.* (2006). The proposed variant, called Comprehensive Learning PSO (CLPSO), generally has a good balance between high and low diversity. Thus, it can cope well with multi-modal functions. However, CLPSO is not the best choice for solving unimodal problems due to its slow convergence (Liang *et al.* 2006).

In this paper, a new adaptive variant of CLPSO is proposed. The proposed method, called Fuzzy-Controlled CLPSO (FC-CLPSO), uses a fuzzy controller to adapt the control parameters of CLPSO. FC-CLPSO is compared with CLPSO and the recent SPSO2011 (available at <http://particleswarm.info>) on 11 benchmark functions.

Section 2 provides an overview of CLPSO. FC-CLPSO is introduced in Section 3. The experimental results are presented and discussed in Section 4. Section 5 concluded the paper.

2 Comprehensive Learning Particle Swarm Optimizer (CLPSO)

The Comprehensive Learning Particle Swarm Optimizer (CLPSO) (Liang *et al.* 2006) addresses the premature convergence problem of the basic PSO by allowing each particle to learn from the personal best position of other particles. Each dimension of a particle can potentially learn from the best experience of a different particle (referred to as an *exemplar*). Hence, the velocity updating equation is modified as follows,

$$v_{i,j}(t+1) = wv_{i,j}(t) + cr_j(t)(p_{f_i(j),j}(t) - x_{i,j}(t)) \quad (3)$$

where $\vec{f}_i = [f_i(1), f_i(2), \dots, f_i(D)]$ and $f_i(j) = i_1$ with $j \in \{1,2, \dots, D\}$ and $i_1 \in \{1,2, \dots, N\}$; \vec{f}_i defines which particle's personal best position particle i should follow. $p_{f_i(j),j}$ can be the corresponding component of any particle's personal best position including the particle's own personal best experience. To generate $p_{f_i(j),j}$, a random number is generated. If this number is greater than a learning probability, Pc_i , this component (i.e. dimension) will learn from its own best experience, otherwise, it will learn from another particle's personal best position. If all exemplars of a particle are its own personal best position, one component will be randomly chosen to learn from another particle's personal best position. If a particle failed to improve itself for m (known as the *refreshing gap*) consecutive iterations, new exemplars will be chosen for that particle. Liang *et al.* (2006) empirically determined that 7 is a good value for m . The learning probabilities are computed using,

$$Pc_i = 0.05 + 0.45 \frac{(\exp(\frac{10(i-1)}{N-1}) - 1)}{(\exp(10) - 1)} \quad (4)$$

According to Liang *et al.* (2006), different learning probabilities affect the exploration/exploitation abilities of the particles.

3 Fuzzy Controlled CLPSO (FC-CLPSO)

In the proposed method, each particle has its own learning probability, inertia weight and acceleration coefficient. A fuzzy controller is used to adapt these three control parameters (i.e. Pc_i , w_i and c_i). The controller takes the normalized rank of each particle as its input and generates three control parameters as outputs. The rank is determined by first sorting the swarm's particles according to their cost function. The best particle (one with the smallest error) is given a rank of 1 (i.e. $R_1 = 1$), while the worst particle (with the biggest error) is given a rank of N (i.e. $R_N = N$). The rank is then normalized to a value between 0 and 1 using,

$$r_i = 1 - \frac{R_i - 1}{N - 1}$$

The normalized ranks are then assigned as membership grades in 3 fuzzy subsets as follows: LOW, MEDIUM and HIGH. The membership functions for Pc_i , w_i and c_i are also defined in a similar way. There are many alternative membership functions that can be used. In this study, a Gaussian curve membership function is chosen for the input and outputs of the fuzzy controller. Figure 1 shows these membership functions.

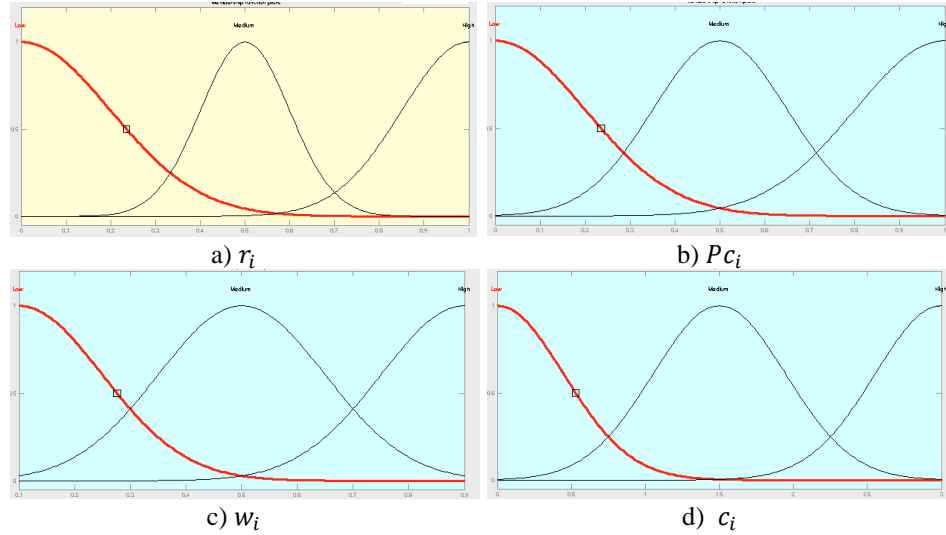


Fig. 1. Membership functions for inputs and outputs.

The fuzzy rules of the fuzzy controller are defined as follows:

IF r_i is LOW **THEN** Pc_i is HIGH AND w_i is HIGH AND c_i is HIGH
IF r_i is MEDIUM **THEN** Pc_i is MEDIUM AND w_i is MEDIUM AND c_i is MEDIUM
IF r_i is HIGH **THEN** Pc_i is LOW AND w_i is LOW AND c_i is LOW

The rationale behind the above rules is that if a particle has a low rank this means it has a low performance, thus, it needs to learn from other particles. This can be done by increasing its Pc_i and c_i . Moreover, such a particle needs to focus more on exploring the search space, hence, its w_i should be increased. On the other hand, best particles should focus more on exploitation (i.e. local search), thus, w_i should be decreased. Moreover, such particles do not need to learn often from other particles, hence, Pc_i and c_i should be decreased.

The fuzzy controller is called whenever new exemplars are need for a particle (i.e. initially and when a particle failed to improve for m iterations).

4 Experimental Results

FC-CLPSO is compared with CLPSO and SPSO 2011. SPSO2011 is generally considered as an outstanding algorithm (Xiang *et al.* 2014). To test the performance of the different methods, 11 functions have been chosen:

1. Six functions from the CEC'2008 Special Session and Competition on large-scale global optimization (Tang *et al.* 2007):
 - (a) Unimodal functions: F1 – Shifted Sphere and F2 – Shifted Schwefel's Problem 2.21.
 - (b) Multimodal functions: F3 – Shifted Rosenbrock, F4 – Shifted Rastrigin, F5 – Shifted Griewank and F6 – Shifted Ackley.
2. Five functions from the ISDA'2009 test suite (Herrera and Lozano 2009): F7 - Schwefel's Problem 2.22, F8 – Schwefel's Problem 1.2, F9 – Extended f10 (Whitley *et al.* 1995), F10 – Bohachevsky and F11 – Schaffer. All functions are unimodal functions.

For all the benchmark functions:

- Number of dimensions is 30.
- Swarm size is 50.
- Number of function evaluations (nfe) is $5000D$.
- The admissible error is $1.00e-4$.
- The number of independent runs is 30.

To measure the *effectiveness* of a method we use two metrics:

- (1) The median of the best-of-run error, which is defined as the absolute difference between the best-of-the-run $f(X^*)$ value and the actual optimum $f(X')$ of a given function.

$$\text{err.} = |f(X^*) - f(X')|$$
- (2) Success rate (SR): The number of successful runs, where a run is successful if $\text{err.} \leq \text{admissible error}$.

All programs are implemented using MATLAB® version 8.1.0.604 (R2013a), and machine epsilon is $2.2204\text{e-}16$. For the pseudo-random number generator (RNG) we have used the `rand` built-in function provided by MATLAB. This function implements the Mersenne-Twister RNG (Matsumoto and Nishimura 1998). We warmed the RNG by calling it 10,000 at the start of the program as suggested by Jones (2010). The non-parametric Friedman's test with $\alpha = 0.05$ and the Dunn-Sidak correction as a post-hoc test have been used to compare the difference in performance of the different algorithms. In this study, the Null Hypothesis, H_0 , states that there is no difference between the *medians* of errors of the different algorithms.

Table 1 shows the median and SR of CLPSO, SPSO2011 and FC-CLPSO on the test functions. The statistically significant best solutions have been shown in **bold**. The results show that FC-CLPSO outperformed CLPSO on 8 functions while performing equally well on the rest. There is no single function where CLPSO performed better than FC-CLPSO. On the other hand, FC-CLPSO outperformed SPSO2011 on 7 functions while SPSO2011 performed better on two functions (i.e. F2 and F8). These two functions are unimodal functions.

The total number of functions solved by FC-CLPSO is 5, while CLPSO and SPSO2011 solved 2 and 4 functions, respectively.

Figure 2 shows the progress of the mean best errors found by CLPSO, SPSO2011 and FC-CLPSO over 30 runs for selected functions. The figure shows that FC-CLPSO reached better solutions faster than the other methods while SPSO2011 prematurely converged on the four functions.

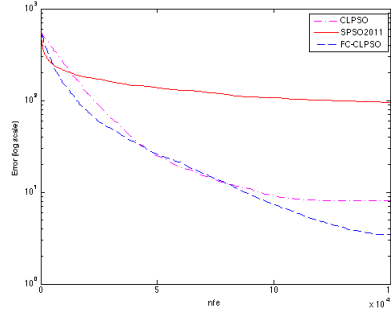
In general, the results clearly shows that using the fuzzy controller to tune CLPSO's parameters improve the performance of CLPSO on most of the benchmark functions.

5 Conclusions

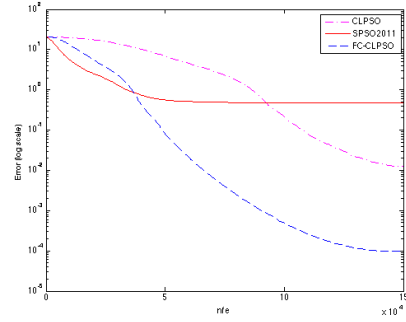
A fuzzy-controlled CLPSO was proposed and compared with CLPSO and SPSO2011. The results showed that using a fuzzy controller to tune the probability learning, inertia weight and acceleration coefficient of each particle significantly improved the performance of CLPSO. Future work will investigate the proposed approach on more benchmark functions and some real-world problems. Moreover, comparison with other *state-of-the-art* methods will be conducted.

Table 1. Comparing CLPSO, SPSO2011 and FC-CLPSO on the test functions.

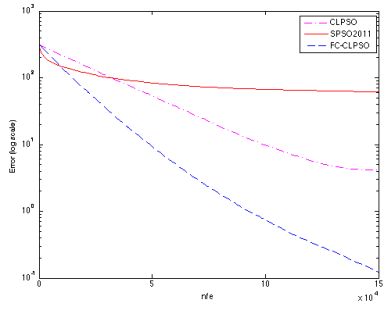
f	CLPSO		SPSO2011		FC-CLPSO		p -value
	<i>Median</i>	<i>SR</i>	<i>Median</i>	<i>SR</i>	<i>Median</i>	<i>SR</i>	
F1	9.64e-05	93.3	9.60e-05	100	9.43e-05	100	0.3932
F2	1.16e+02	0	1.97e-01	0	7.79e+01	0	9.3576e-14
F3	1.89e+02	0	4.81e+02	0	1.15e+02	0	4.1186e-06
F4	8.50e+00	0	9.89e+01	0	2.99e+00	0	2.4603e-13
F5	7.48e-04	0	9.98e-05	53.3	9.75e-05	90	3.3633e-05
F6	1.17e-02	0	9.97e-05	66.7	9.84e-05	100	6.6018e-08
F7	1.75e-04	0	1.17e+00	0	9.80e-05	100	9.3576e-14
F8	1.86e+03	0	8.58e+01	0	1.99e+02	0	1.3924e-12
F9	4.13e+00	0	6.02e+01	0	1.44e-01	0	9.3576e-14
F10	8.38e-05	100	9.40e-05	80	8.74e-05	100	0.0718
F11	3.98e+00	0	6.47e+01	0	1.02e-01	0	9.3576e-14



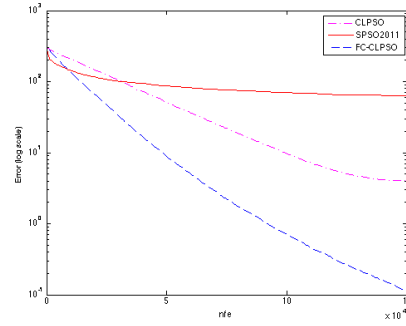
(a) F4



(b) F6



(c) F9



(d) F11

Fig. 2. Mean best error curves of CLPSO, SPSO2011 and FC-CLPSO for selected functions.

6 References

- R. Eberhart and J. Kennedy. A New Optimizer Using Particle Swarm Theory. In: 6th International Symposium on Micromachine and Human Science, pp. 39-43, IEEE Service Center, Piscataway, NJ, 1995.
- F. Herrera and M. Lozano. Benchmark functions 7-11. Technical Report, Workshop: Evolutionary Algorithms and other Metaheuristics for Continuous Optimization Problems - A Scalability Test, <http://sci2s.ugr.es/programacion/workshop/functions7-11.pdf>, 2009.
- D. Jones. Good practice in (pseudo) random number generation for bioinformatics applications. Technical report, UCL Bioinformatics Group, 2010.

J. Liang, A. Qin, P. Suganthan and S. Baskar. Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. *Transactions on Evolutionary Computation*, vol. 10(3), pp. 281-295, 2006.

M. Matsumoto and T. Nishimura. Mersenne Twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, vol. 8(1), pp. 3-30, 1998.

A. Olsson. *Particle Swarm Optimization: Theory, Techniques and Applications*. Nova Science Pub Inc., 2011.

J. Riget and J. Vesterstrøm. A Diversity-Guided Particle Swarm Optimizer - the ARPSO. Technical report, EVALife, Denmark, 2002.

K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, and Z. Yang. Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization. Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2007.

D. Whitley, R. Beveridge, C. Graves and K. Mathias K. Test driving three 1995 genetic algorithms: New test functions and geometric matching. *Journal of Heuristics*, vol. 1(1), pp.77–104, 1995.

Y. Xiang, Y. Peng, Y. Zhong, Z. Chen, X. Lu and X. Zhong. A particle swarm inspired multi-elitist artificial bee colony algorithm for real-parameter optimization. *Computational Optimization and Applications*, vol. 57, pp. 493-516, 2014.