Undefined 1 (2013) 1–12
IOS Press

The Multi-Objective Hybridization of Particle Swarm Optimization and Fuzzy Ant Colony Optimization

Walid Elloumi [a,*], Nesrine Baklouti [a] Ajith Abraham [b,c] and Adel M. Alimi [a]
[a] *REGIM: REsearch Groups on Intelligent Machines, University of Sfax, National Engineering School of Sfax (ENIS), BP 1173, Sfax, 3038, Tunisia*
E-mail: {walid.elloumi,nesrine.baklouti,adel.alimi}@ieee.org
[b] *MIR Labs: Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, WA, USA*
[c] *IT4Innovations - Center of excellence, VSB - Technical University of Ostrava, Czech Republic*
E-mail: {ajith.abraham}@ieee.org

Abstract. In this paper, we illustrate a novel optimization approach based on Multi-objective Particle Swarm Optimization (MOPSO) and Fuzzy Ant Colony Optimization (FACO). The basic idea is to combine these two techniques using the best particle of the Fuzzy Ant algorithm and integrate it as the best local Particle Swarm Optimization (PSO), to formulate a new approach called hybrid MOPSO with FACO (H-MOPSO-FACO). This hybridization solves the multi-objective problem, which relies on two time performance criteria and the shortest path. Experimental results illustrate that the proposed method is efficient.

Keywords: Multi-objective, Particle Swarm Optimization, Ant Colony Optimization, Fuzzy, Swarm Intelligence

1. Introduction

In many areas of science, the optimization procedure often has more than one objective; hence, the need for multi-objective optimization is conspicuous. One of the factors that differentiate single objective optimization with respect to the multi-objective optimization is the fact that the optimal solution for multi-objective optimization is not necessarily unique.

In general, the multi-objective version of a problem is more difficult than the single case of the goal. In a typical problem of multi-objective optimization (also known as multi-criteria optimization) there is a family of equivalent solutions that are superior to the rest of the solutions which are considered equal in terms of the simultaneous optimization of several objectives (and possibly competing) functions [1].

Within the multi-objective optimization there is no single solution. Instead, the interaction of multiple targets gives an effective set of solutions (non-inferiority) or non-dominated solutions known as the Pareto-optimal, which gives the decision maker a greater flexibility in the choice of a suitable alternative [2].

In other words, the multi-objective optimizer is expected to give a set of all equivalents, diverse and representative solutions. Objectives to optimize simultaneously can be mutually contradictory. In addition, the implementation of appropriate diversity in the solutions while approaching convergence is another challenge in multi-objective optimization [3].

Different evolutionary algorithms and swarm intelligence approaches have been validated for multi-objective optimization [5,36,42] problems. Evolution-

*Corresponding author. E-mail: walid.elloumi@ieee.org

ary algorithms and swarm intelligence approaches usually do not provide nor identify optimal compromise, but they try to find a good approximation, i.e. a set of solutions whose objective vectors are (thankfully) not too far from optimal vectors objective [4,44,50].

Miscellaneous multi-objective evolutionary approaches [6] and swarm intelligence approaches are available, and we are certainly interested in the technique that brings about the best approximation for a given problem [7,8,21,45,47]. In recent years, there has been an increased interest in the study, design and analysis of particle swarm optimization approaches to solve the problems of multi-objective optimization interest. Thanks to its fast convergence [38], PSO has been advocated to be particularly suitable for multi-objective optimization.

PSO is a population-based field of swarm intelligence approach which was originally developed by James Kennedy and Russell Eberhart [9]. Their idea was to simulate the social behavior of a flock of birds trying to reach an unknown destination (depending on fitness), for example, the location of food resources via flying across the field (space research) [32,48]. In other words, PSO is inspired by the adaptation of a natural system based on the metaphor of communication and social interaction. Despite its simplicity, PSO offers several effective and specific solutions but it has also many problems of multi-objective engineering [6,7,40].

In this research, we proposed a hybrid approach MOPSO with FACO (Multi-objective Particle Swarm Optimization (MOPSO) and Fuzzy Ant Colony Optimization (FACO)). We proposed a new modified fuzzy ant colony optimization technique. The fuzzy notion allocates the ants some intelligence in following the pheromone. We choose the use of fuzzy logic (FL) because of its efficiency in modeling data and control applications [33]. It was initially proposed by Zadeh [26], and used in several applications [27,28,41].

In the literature, ACO based fuzzy control systems have been proposed in several works [22,23,24,25], but those works are essentially for tuning fuzzy systems with ACO algorithm. The notion of a modified ACO with fuzzy or simply the fuzzy ant only appeared in few works [43,49]. In [18] and [19], authors presented ant fuzzy based fuzzy approaches for a Clusterization problem. In [20], authors controlled a ball and beam system with a fuzzy ant approach, where they controlled the level of pheromone update using a fuzzy logic system. The big difference between this work and ours consists essentially in applying multi-objective particle swarm optimization hydride with fuzzy ant colony optimization [29].

The rest of the paper is organized in five sections. In the Section2, we give an overview of decision groups using Fuzzy ant and particle swarm optimization in general. Different problems solving our hybrid approach MOPSO with FACO inspired by the collective behavior of insect colonies and flock of birds that are introduced using computational swarm intelligence are proposed in Section 3. Section 4 presents the comparative study and experimental results and finally the paper concludes in Section 5.

2. Decision Groups

Decision making requires multiple perspectives of different people as a decision-maker may not have enough knowledge to properly solve a single problem. This is especially true when the decision environment becomes more complex. Therefore, organizational decisions are taken in groups like Ant Colony and Swarm particles [30,39].

These decisions may be the product design, and the development of policies and strategies, employee selection, and the organization of various resources. In an organization, a group decision is self-regulation, task-oriented work as an independent committee group. Group-based decision making has become an essential element for the proper functioning of an organization [10].

Decision-making group is the process of arriving at a judgment or a solution to a decision problem based on the input and the return of several people as PSO and fuzzy ant colony. This group work can achieve a satisfactory solution for the group rather than the best solution because it hardly exists. In general, a satisfactory solution of the group is the one that is more acceptable to the group of people as a whole. Since the impact of the selection of the appropriate solution affects organizational performance, it is important to do the decision-making process as efficient as possible. It is, therefore, compulsory to determine what makes an effective solution, and to increase the overall level of satisfaction of the solution through the group.

2.1. *PSO and MOPSO Approches*

PSO is basically a technique of parallel multi-agent research. PSO consists of three stages, namely, the positions and velocities of particle production, updating speed, and finally, updating position. PSO is easy to implement in computer simulations using mathematical operations and basic logic, since its working mechanism involves only two basic rules update.

The particles are conceptual entities that constitute a swarm flying through space multidimensional search. The relationship between the swarm of particles in PSO is similar to the relationship between population and chromosomes in the genetic algorithm [37,46]. At any given time, each particle has a position and a velocity.

The position vector of a particle relative to the origin of the search space is a test solution to the problem of search. These particles fly with a certain velocity and find the best global position after some iteration. At each iteration, each particle can adjust its velocity vector, based on its momentum and the influence of his best position (pbest - particle best) and the best position of its neighbors (gbest - Global Best), then calculate a new position that "particle" is to fly. In other words, it finds the global optimum by simply adjusting the trajectory of each individual towards its own best location and the best particle swarm with each generation of evolution [34].

The direction of the particle swarm is defined by the set of neighboring particles of the particle and its historical experience. The manuscript [31] proposed the first extension of the PSO strategy for solving multi-objective problems. There have been several recent proposals using basic PSO to handle multiple objectives, surveyed in [6]. However, the speed of convergence of MOPSO approaches often implies a rapid loss of diversity during the optimization process. In this context, several MOPSO have difficulty controlling the balance between exploration and exploitation.

In [11], the authors propose a multi-objective PSO (MOPSO) incorporating the concept of density estimator the nearest neighbor to both select the best global particle and remove particles from the external archive of non-solutions dominated. When selecting a leader, the archives of non-dominated solutions is sorted in descending order with respect to the density estimator, and a particle is chosen at random from the top of the list. On the other hand, when the external archive is full, it is sorted in descending order with respect to the value of the density estimator and a new particle is randomly chosen to remove, from the bottom of the list.

This approach uses the mutation operator proposed in [12] so that it is applied only for a certain number of generations, in the early stages.

$$v_{ij}(t+1) = w * v_{ij}(t) + c_1 * r_1(p_{il}(t) - x_{ij}(t)) + c_2 * r_2(p_{ig}(t) - x_{ij}(t)) \tag{1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \qquad 1 \leq i \leq K \tag{2}$$

Where the inertia weight w , c_1 and c_2 are constants, r_1 and r_2 are random variables in the range from 0 to 1. $p_{il}(t)$ is the best local solution of the i th particle for the iteration number up to the i th iteration and the $p_{ig}(t)$ is the best global solution of all particles. A particle velocity should be updated using Eq. (1), the particle is moved according to Eq. (2).

2.2. Some important ACO variants

In this model, c ants per second across the bridge in each direction at a constant speed cm/s, the filing of a unit of pheromone on the branch. Given the length l_s and l_l $(incm)$ of the short and the long arm, an ant choosing the short length cross in $t_s = l_s/v$ seconds, while the ant choice of the long arm use $r * t_s$ seconds where $r = r = l_l/l_s$. The probability $p_{ij}(t)$ an ant arriving at the decision point $i \in \{1,2\}$, selects branch $j \in \{s,l\}$, where S and L denote the short and long arm, respectively, the time t is set to be based on the total amount of pheromone $\tau_{ij}(t)$ on the branch, which is proportional to the number of ants that used the branch until time t, $\tau_{ij} = 1/d_{ij}$ which are called heuristic information and are two parameters that control the relative importance of the intensity of the track. For example, the probability $p_{ij}(t)$ to choose the short leg is given by :

$$P_{ij}^k(t) = \begin{cases} \dfrac{(\tau_{ij}(t)^\alpha)(\eta_{ij})^\beta)}{\sum_{i \in J_i^k}(\tau_{ij}(t)^\alpha)(\eta_{ij})^\beta} & \text{Si } (i,j) \in J_i^k \\ 0 & \text{Si } (i,j) \notin J_i^k \end{cases} \tag{3}$$

We deduce that after soon found a path between Home and Food, each ant leaves certain quantities of pheromone $\Delta_{ij}^k(t)$ which depends on:

$$\Delta_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L^k(t)} \text{ Si } (i,j) \in T^k(t) \\ 0 \quad \text{Si } (i,j) \notin T^k(t) \end{cases} \tag{4}$$

Where $T^k(t)$ is the target taken by the ant k for iteration t, $L^k(t)$ is the length between Home and Food and Q is the fixed parameter.

Now, we indicate variants to the best-known and frequently used Fuzzy ACO algorithms. It should be mentioned that these features represent only the nuclei of algorithms based on ant. The ant system (AS) is the first ACO algorithm which has been proposed in the literature [13]. It is characterized by the following rule for updating the pheromone which ρ refers to the rate of pheromone evaporation.

$$\tau_{ij} = (1-\rho).\tau_{ij} + \frac{\rho}{K}\sum_{i=1}^{K}\Delta\tau_{ij}^{k} \tag{5}$$

With $\rho \in [0,1]$ is called evaporation rate, and noted by w^k path of the agent k.

We notice that within the ant system, all employees contribute to the increment of the amount of pheromone and this increment is chosen as a proportional shape. The Ant System (AS) MIN-MAX (MMAS), was developed by Stutzle and Hoos [14,15]. It is mainly characterized by two innovations compared to the ant system: first, instead of allowing all agents to deposit the amount of pheromone on their way, we are only interested in paths that are proven or strengthened. Two selection rules are used alternately or in combination. The first one has been the best so far (BS: best-so-far) the best path found so far is strengthened, the other is the best iteration (IB iteration best), where the best path has been found in the current iteration is reinforced. Second, to recover the amount of pheromone caused by the restriction reinforcement only "exceptional" paths MMAS apply the lower and upper limits for the pheromone trails. There are two additional changes compared with AS suggested by [2,3], about a special way to initialize pheromone and can reset pheromone trails when stagnation occurs. For a fixed iteration k, we note that w is the best path found in iterations $(1, ..., k)$. The best so far w' path is updated each time we strictly improve the current path $\hat{w}$ is found. Then the pheromone update is given by:

$$\tau_{ij} = (1-\rho).\tau_{ij} + \frac{\rho}{K}\sum_{i=1}^{K}\Delta\tau_{ij}^{best} \tag{6}$$

With $\hat{w} = w^{best}$ and $w^{best} = w'$ in this case the choice of the best so far and the best iteration is respectively identical. We refer to both cases by MMASbs Abbreviations MMASib respectively. In this section, we will not address the combined cases. To allow a simpler situation, we also consider the case where the reward for reinforced edges on paths is not chosen in a proportional form.

2.3. The modified fuzzy ant algorithm

For updating the pheromone quantity, we proposed a modified fuzzy ant colony optimization technique which is based on the min and max pheromone information. The fuzzy notion allocates the ants some intelligence in following pheromones. As we seen the pheromone is updated within Eq. (6). To ameliorate the speed and the convergence of the ant we propose in this paper a fuzzy procedure of the update of the pheromone. The fuzzy use can orient ant to a shorter path with the suitable level of pheromone quantity. In the conception of our fuzzy logic system we need to conceive three parts the fuzzification, the inference and the defuzzification.

2.3.1. Fuzzification

This part consists in defining inputs and outputs of our system under a linguistic form. The Input of our fuzzy system is $\dfrac{Q}{L^k(t)}$, which we choose to subdivide it into five membership functions: VeryLow, Low Medium, Important and VeryImportant as illustrated below:

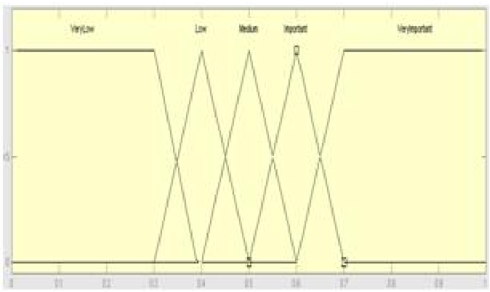

Fig. 1. Input of the fuzzy ant system

Whereas the system output which is the fuzzy output pheromone $\sum_{i=1}^{K} \Delta\tau_{ij}^{best}$ is represented by four membership functions Zero, Short, Medium and Long are as illustrated in Figure 2:

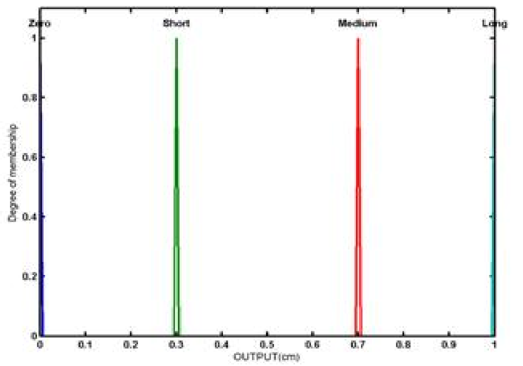

Fig. 2. Output of the fuzzy ant system

2.3.2. Inference

The inference corresponds to the definition of If-Then rules describing the system working. In our fuzzy ant system, we conceived five rules for describing the adequate quantity of pheromone to be taken desponding on the value of $\dfrac{Q}{L^k(t)}$ which are represented as follows:

R1: IF $\dfrac{Q}{L^k(t)}$ IS VeryLow THEN $\sum_{i=1}^{K} \Delta\tau_{ij}^{best}$ is Zero

R2: IF $\dfrac{Q}{L^k(t)}$ IS Low THEN $\sum_{i=1}^{K} \Delta\tau_{ij}^{best}$ is Long

R3: IF $\dfrac{Q}{L^k(t)}$ IS Medium THEN $\sum_{i=1}^{K} \Delta\tau_{ij}^{best}$ is Medium

R4: IF $\dfrac{Q}{L^k(t)}$ IS Important THEN $\sum_{i=1}^{K} \Delta\tau_{ij}^{best}$ is Short

R5: IF $\dfrac{Q}{L^k(t)}$ IS Very Important THEN $\sum_{i=1}^{K} \Delta\tau_{ij}^{best}$ is Zero

For example $\nu_{VeryLow} \dfrac{Q}{L^k(t)}$ expresses the degree of membership function of the input $\dfrac{Q}{L^k(t)}$ in the membership function $VeryLow$. In the inference part we used the min-max mamdani method, and we search the degree of activation of each rule using the degree of membership function of the input $\nu_A \dfrac{Q}{L^k(t)}$ in the appropriate membership function.

2.3.3. Defuzzification

The defuzzification part corresponds in computing outputs command. In our system, we used the centroid of sets which is the most used method in literature that is computed using the following equation to retrieve $\sum_{i=1}^{K} \Delta\tau_{ij}^{best}$:

$$\sum_{i=1}^{K} \Delta\tau_{ij}^{best} = \frac{\sum_{i=1}^{5} \frac{Q}{L^k(t)} X \nu_A \frac{Q}{L^k(t)}}{\sum_{i=1}^{5} \nu_A \frac{Q}{L^k(t)}} \tag{7}$$

3. Our Hybrid Approach MOPSO with FACO

As we begin to initialize the settings of Fuzzy Ant Colony Optimization (FACO) and Multi Objective Particle Swarm Optimization (MOPSO) in parallel thereafter, we have made the operation of Fuzzy Ant ie to move in space Research using the pheromone deposited by moving and the probability to move from one state to another. If Fuzzy Ant encounters an obstacle, then it has four choices either to move up, down, left or right as appropriate. We safeguard the operation of the Fuzzy Ant and integrate as local particle for MOPSO. Thereafter, we vary the speed and position of the MOPSO.

As soon as it comes from the nest to food, ie we could find a way, it saves it in a table to facilitate the return of food to the nest that is to say, the scheme return. This hybridization between FACO and MPSO best particle using the Fuzzy Ant as the best local particle called expired H-MOPSO-FACO has helped us win two criteria: the first time in the seconds and the second is the shortest way. This is understandable because MOPSO encounter an obstacle. They will not meet all directly that is to say they will not confront the obstacle, because the information has Fuzzy Ant will turn to the particle MOPSO, which saves us at the time and shortest path. These are the two essential criteria to minimize our problem. Especially the Fuzzy Ant is self organized while MOPSO are socio-cognitive, and we could do this hybridization which allowed us to overcome the major handicap of MOPSO because the PSO represents a weak point at save all particles, but cons are faster than the Fuzzy Ant. Therefore we have used the amount of pheromone deposited by Fuzzy ant as a memory for PSO.

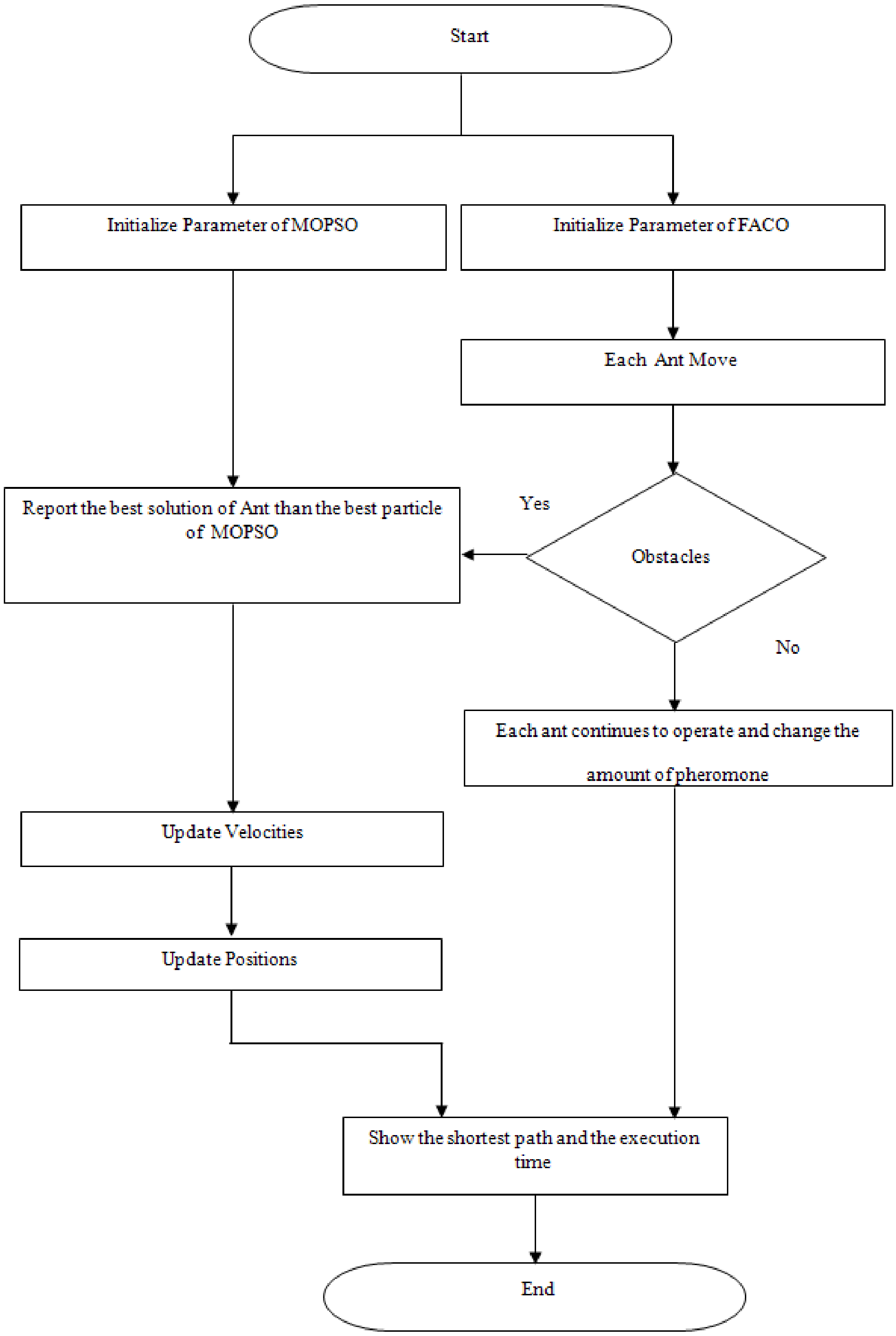

Fig. 3. Hybridized approach

4. Experimental results

In this section, we describe the results obtained with the same configuration problem that is the path of the path from the nest to the food using obstacles.

4.1. Evaluation of the proposed MOPSO

For evaluating our proposed MOPSO, we chose our multi-objective fitness function as:

$$f(x, y) = cos(2*pi*x)*cos(2*pi*y)*e^{((x^2+y^2)/10)} \quad (8)$$

The experiments were conducted for 20 independent runs in MATLAB environment to evaluate the performance of MOPSO. The adopted setup for the MOPSO was $c_1 = c_2 = 1.2$, and the range of the inertia weight w is from 0.5 during the generations for the MOPSO approaches. The population size was 50 particles, stopping criterion, t_{max}, of 100 generations.

Simulation results are presented in Figures 4-7. Figure 4 depicts the initialization of the particle swarm to f(x, y) function in three dimensions. Black points in the designed Figure 4 refer to the initialized particles.

As Figure 5 illustrates the behavior of swarm movement after it was noted that we can not show all the best results which depicts the efficiency of the proposed hybridization scheme. Figure 6 depicts the movement of the behavior of swarms hybridized with Fuzzy ant colony, which shows us a better solution compared to what exists. Figure 7 shows the rapid convergence of the global best particle according to the number of iterations.

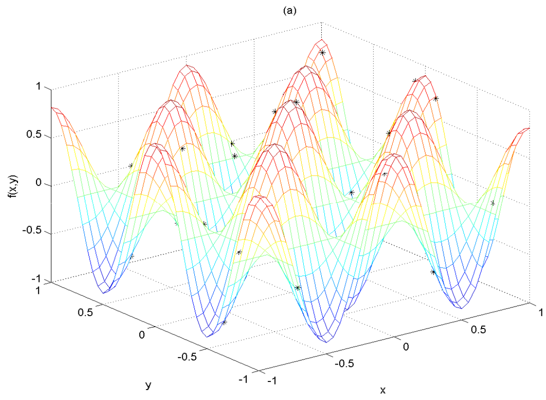

Fig. 4. Representation of initial function

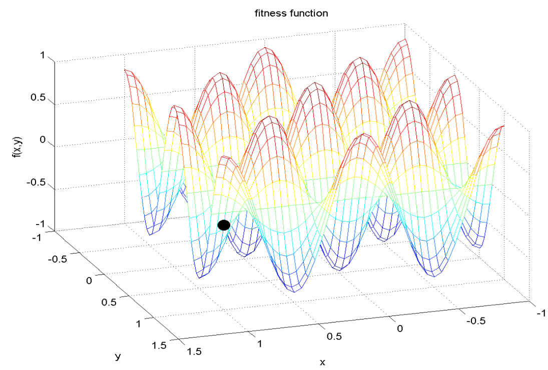

Fig. 5. Representation of the function after the motion of particles (MOPSO)

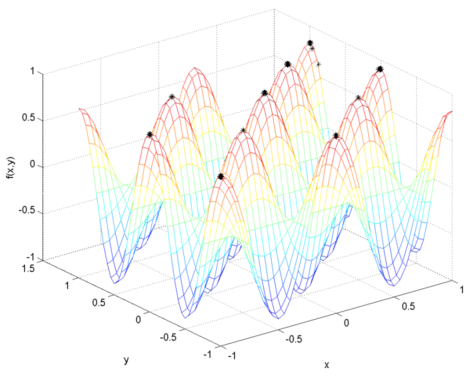

Fig. 6. Representation of the function after the motion of particles (H-MOPSO-FACO)

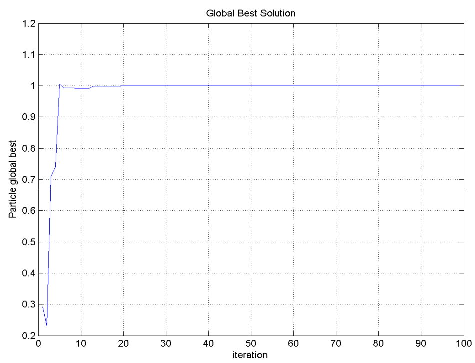

Fig. 7. Convergence of global best particle

4.2. Evaluation of the proposed Hybrid MOPSO with FACO

Our approach H-MOPSO-FACO is coded in C and run on a Pentium(R) Dual core CPU 2 GHz PC with 2.92Go memory. We have proposed an Ant colony algorithms supervised by Particle Swarm Optimization to solve continuous optimization problems [35]. Traditional ACO is used for discrete optimization while PSO is for continuous optimization problems. Together PSO and ACO have shown a great potential in solving a wide range of optimization problems. All results have been presented in the paper [17].

We have represented the particles as + and it moves from the nest to food. Once a particle reaches food we mark it on the brand as Ö.

There are many parameters used for the H-MOPSO-FACO: The Size of population=50 (swarm: ant and bird), having C_1 and C_2, set as $C_1 = C_2 = 2$. While the inertia weight w is taken as 0.9, and the maximum of velocity v is taken as 100 and dimension of space as 10. Both α and β control the relative significance of pheromone trail and distance, where $\alpha = 1.5$, $\beta = 2$. ρ refers to the rate of pheromone evaporation which is taken $\rho = 0.3$.

Table 1 summarizes the new results improved regarding what was published in our previous article [16] about the combination and comparison between H-MOPSO-FACO. This table shows us the different results between the different methods used. First we began by testing the method of fuzzy Ant using its specific rules. Thereafter we tried to use or apply the same problem of seeking the shortest path between the nest which is represented by H and F in food by using our GUI obstacles. For the same problem, we tested our hybridization method between multi-objective Particle Swarm Optimization and Fuzzy Ant Colony (FACO). For the considered methods we applied two evaluation criteria; the execution time and the route of the shortest path.

Figure 8 shows the simulation of MOPSO does not adequately provide the shortest route proving that the MOPSO is low in memory and shows that hybridization is necessary to improve the performance.

Undefined 1 (2013) 1–12
IOS Press

Table 1

Comparison between the different methods

Method Name	Time with obstacles	Best length with obstacles	Without obstacles
Fuzzy Ant	Between 3 and 5 minutes	it shows between the Home and Food	same time and shortest path
MOPSO	Between 1 and 3 minutes	we can not see between the Home and Food	same time and shortest path
H-MOPSO-FACO	between a few seconds and 2 minutes	it shows between the Home and Food	same time and shortest path

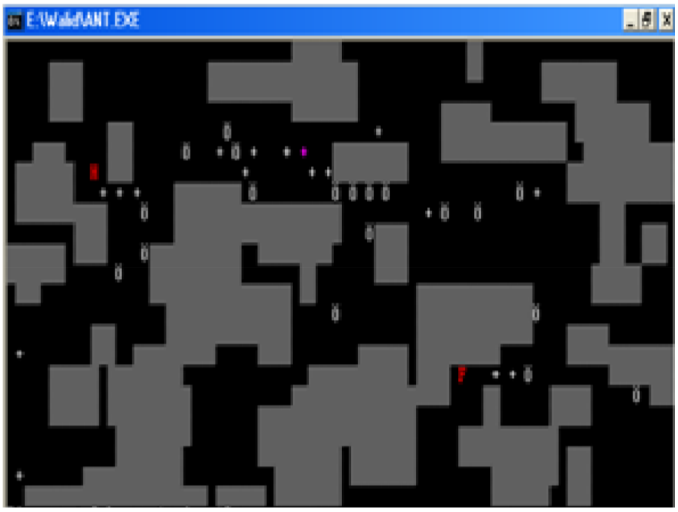

Fig. 8. Simulation with MOPSO

It was found that the MOPSO method is faster compared to the fuzzy Ant, but the criterion for the shortest path cannot visualize the method for MOPSO. MOPSO does not allow us to visualize the shortest path, which explains that the MOPSO is low in memory, and for this we must achieve a combination of MOPSO and FACO that allows us to overcome this major handicap of MOPSO. Table 1 illustrates that the proposed hybridization approach is better than the two other algorithms namely Fuzzy Ant and MOPSO. In fact, we notice that, as the Size of population increases, the time of execution of the hybrid method decreases when compared with the time of Fuzzy Ant and the time of MOPSO. Besides, as the size of population increases more than we have attempted; the distance becomes shorter using the proposed hybrid method.

Results are represented as follows in Figure 9, in which Figures (a)-(f) represents different cases of our algorithm with obstacles.

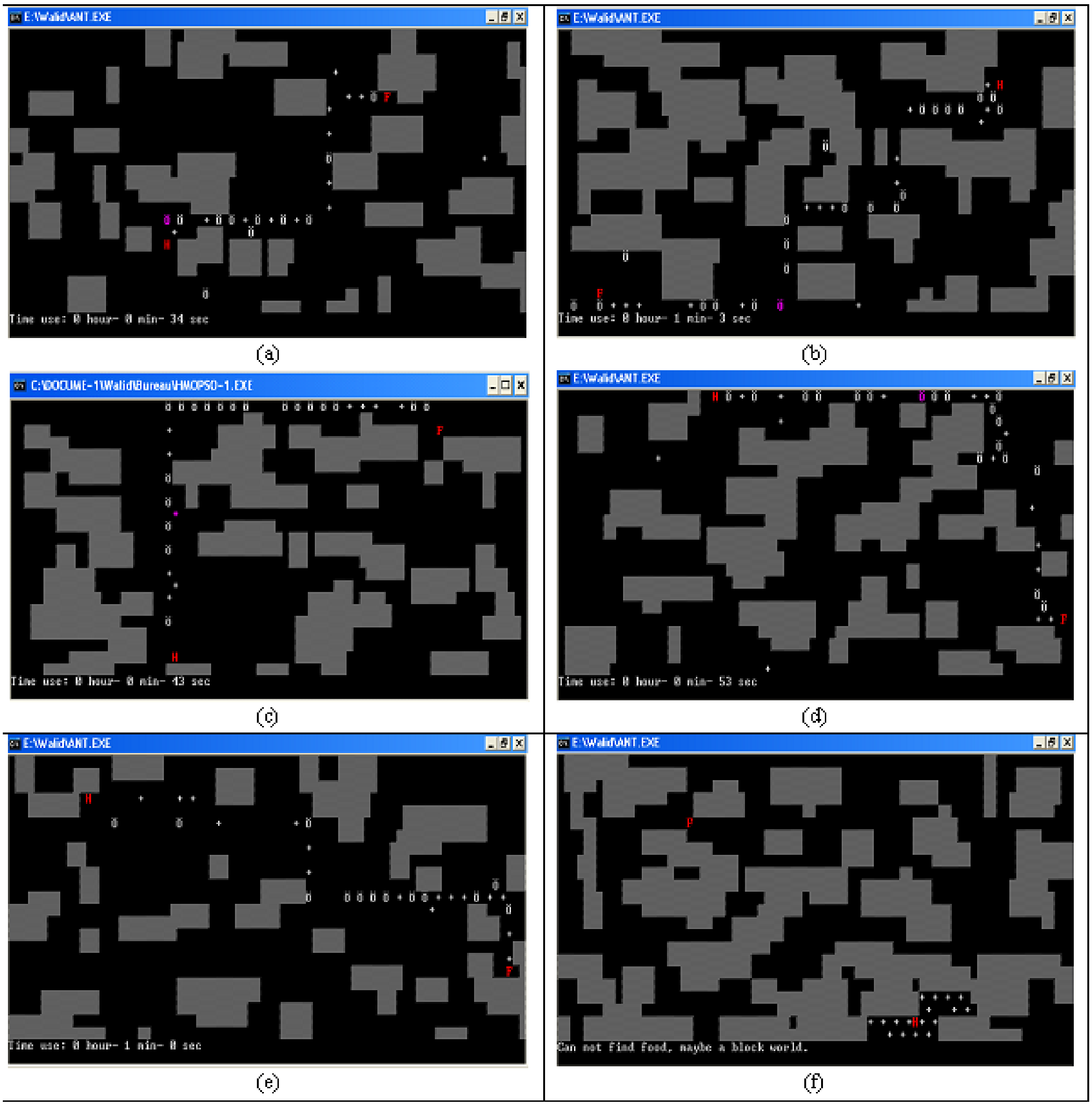

Fig. 9. Simulation results

5. Conclusions

In this paper, we illustrated the operation of the PSO, MOPSO, Fuzzy Ant Colony and the hybridization between these techniques in order to minimize the execution time and the route of the shortest path, which we formulated as a multi-objective optimization problem. The proposed hybridization approach has saved us the computational time and more better results, which is not the case when using Ant colony alone. The future work is to apply the hybridization to incorporate the difference between the particles into PSO and to vary the inertia weight according to the number of particles. For every particle, the fitness of its personal best is considered as an input to the fuzzy system for calculating the variation of its inertia weight.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

References

[1] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca and V. G. Fonseca, *Performance Assessment of Multi-objective Optimizers*, IEEE Transactions on Evolutionary Computation, p 117-132, 2003.

[2] D. A. Van Veldhuizen and G. B. Lamont, *Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art*, Evolutionary Computation, p 125-147, 2000.

[3] C. Coello, *A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization*, Knowledge and Information Systems, p 269-308, 1999.

[4] C. Coello, D. A. Van Veldhuizenand and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, New York, 2002.

[5] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, Interscience Series in Systems and Optimization. John Wiley and Sons, Chichester, 2001.

[6] M. R. Sierra and C. Coello, *Multi-objective Particle Swarm Optimizers: A Survey of the State-of-the-art*, International Journal of Computational Intelligence Research, p 287-308, 2006.

[7] D. Liu, K. C. Tan, C. K. Goh and W. K. Ho, *A Multiobjective Memetic Algorithm Based on Particle Swarm Optimization*, IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics,p 42-50, 2007.

[8] M. A. Abido, *Multiobjective Particle Swarm Optimization for Environmental/ Economic Dispatch Problem*, Electric Power Systems Research, p 1105-1113, 2009.

[9] R. C. Eberhart and J. Kennedy, *A New Optimizer Using Particle Swarm Theory*, In: Proceedings of International Symposium on Micro Machine and Human Science, p 39-43, 1995.

[10] M. Ali, M. Pant and A. Abraham, *Unconventional initialization methods for differential evolution*, Applied Mathematics and Computation, p 4474-4494, 2013.

[11] C. R. Raquel and P. C. Naval Jr, *An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization*, In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2005), Washington, DC, USA, 2005.

[12] C. Coello, G. T. Pulido and M. S. Lechuga, *Handling Multiple Objectives with Particle Swarm Optimization*, IEEE Transactions on Evolutionary Computation, p256-279, 2004.

[13] M. Dorigo, V. Maniezzo and A. Colorni, *Positive feedback as a search strategy*, Technical Report TR-POLI-91-016, Dipartimento di Elletronica, Politecnico di Milano, Milan, Italy, 1991.

[14] T. Stützle and H. H. Hoos, *MAX-MIN Ant System and local search for the travelling salesman problem*, In T. Baeck, Z. Michalewicz and X. Yao (Eds.), Proceedings ICEC '97 international conference on evolutionary computation, p 309-314, 1997.

[15] T. Stützle and H. H. Hoos, *MAX-MIN Ant system*, Future Generation Computer Systems, p 889-914, 2000.

[16] W. Elloumi and A. M. Alimi, *Combinatory Optimization of ACO and PSO*, International Conference on Metaheuristique and Nature Inspired Computing, p 1-8, 2008.

[17] W. Elloumi, N. Rokbani and A. M. Alimi, *Ant Supervised By PSO* International Symposium on Computational Intelligence and Intelligent Informatics, p 161-166, 2009.

[18] E. Lefever, T. Fayruzov, V. Hoste and M. D. Cock, *Clustering web people search results using fuzzy ants* Information Sciences, p 3192-3209, 2010.

[19] S. Schockaert, M. D. Cock, C. Cornelis and E. Kerre, *Clustering web search results using fuzzy ants*, International Journal of Intelligent Systems, p 455-474, 2007.

[20] Y. H. Chang, C. W. Chang, C. W. Tao, H. W. Lin and J. S. Taur, *Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization*, Expert Systems with Applications, p 3624-3633, 2012.

[21] W. Elloumi and A. M. Alimi, *A More Efficient MOPSO for Optimization*, The eight ACS/IEEE International Conference on Computer Systems and Applications, p 1-7, 2010.

[22] H. Boubertakh, M. Tadjine, P. Y. Glorennec and S. Labiod, *Tuning fuzzy PID controllers using ant colony optimization*, Control and Automation, MED '09. 17th Mediterranean Conference p.13-18, 2009.

[23] C. F. Juang and P. H. Chang, *Designing Fuzzy-Rule-Based Systems Using Continuous Ant-Colony Optimization*, Fuzzy Systems, IEEE Transactions, p 138-149, 2010.

[24] M. Dorigo, V. Maniezzo and A. Colorni, *Ant system: optimization by a colony of cooperating agents*, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, p 29-41, 1996.

[25] A. Kumar, M. Hanmandlu and H. M. Gupta, *Ant colony optimization based fuzzy binary decision tree for bimodal hand knuckle verification system*, Expert Systems with Applications, p 439-449, 2013.

[26] L. A. Zadeh. Fuzzy Sets, *Information and Control*, p 338-353, 1965.

[27] N. Baklouti and A. M. Alimi, *Motion Planning in Dynamic and Unknown Environment Using an Interval Type-2 TSK Fuzzy Logic Controller*, In FUZZ-IEEE, IEEE International Conference on Fuzzy Systems, p 1-6, 2007.

[28] I. Kallel, N. Baklouti and A. M. Alimi, *Accuracy Preserving Interpretability with Hybrid Hierarchical Genetic Fuzzy Modeling: Case of Motion Planning Robot Controller*, In Evolving Fuzzy Systems, on International Symposium, p 312-317, 2006.

[29] N. Baklouti and A. M. Alimi, *Type-2 Fuzzy Logic Control of Mobile Robots*, Journal of Intelligent Learning Systems and Applications (JILSA), p 291-302, 2012.

[30] S. S. Kim, J. H. Byeon, H. Liu, A. Abraham and S. F. McLoone, *Optimal job scheduling in grid computing using efficient binary artificial bee colony optimization*, Soft Computing, p 867-882, 2013.

[31] B. Iglesia, A. Reynolds and V. J. R. Smith, *Developments on a multi-objective metaheuristic (momh) algorithm for finding interesting sets of classification rules*, In: C. A. C. Coello, A. Hernt'andez Aguirre and E. Zitzler (eds.) EMO 2005, p. 826-840. Springer, Heidelberg, 2005.

[32] B. Yue, H. Liu and A. Abraham, *Dynamic Trajectory and Convergence Analysis of Swarm Algorithm*, Computing and Informatics, p 371-392, 2012.

[33] C. L. Huang, W. C. Huang, H. Y. Chang, Y. C. Yeh and C. Y. Tsai *Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering*, Applied Soft Computing, p 3864-3872, 2013.

[34] M. S. Kiran, M. Gündüz and Ö. K. Baykan *A novel hybrid algorithm based on particle swarm and ant colony optimization for finding the global minimum*, Applied Mathematics and Computation, p 1515-1521, 2012.

[35] P. S. Shelokar, P. Siarry, V. K. Jayaraman, B. D. Kulkarni *Particle swarm and ant colony algorithms hybridized for improved continuous optimization*, Applied Mathematics and Computation, p 129-142, 2007.

[36] Y. Marinakis, M. Marinaki, M. Doumpos and C. Zopounidis *Ant colony and particle swarm optimization for financial classification problems*, Expert Systems with Applications, p 10604-10611, 2009.

[37] S. M. Chen and C. Y. Chien *Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques* Expert Systems with Applications, p 14439-14450, 2011.

[38] A. Kaveh and S. Talatahari *Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures*, Computers and Structures, p 267-283, 2009.

[39] M. S. Kiran, E. Özceylan, M. Gündüz and Turan Paksoy *A novel hybrid approach based on Particle Swarm Optimization and Ant Colony Algorithm to forecast energy demand of Turkey*, Energy Conversion and Management, p 75-83 2012.

[40] R. Thangaraj, M. Pant, A. Abraham and P. Bouvry *Particle Swarm Optimization: Hybridization Perspectives and Experimental Illustrations*, Applied Maths and Computation, Elsevier Science, Netherlands, p 5208-5226, 2011.

[41] H. Liu and A. Abraham *An Hybrid Fuzzy Variable Neighborhood Particle Swarm Optimization Algorithm for Solving Quadratic Assignment Problems*, Journal of Universal Computer Science, p 1032-1054, 2007.

[42] H. Liu, A. Abraham and M. Clerc, Chaotic *Dynamic Characteristics in Swarm Intelligence*, Applied Soft Computing Journal, Elsevier Science, p 1019-1026, 2007.

[43] H. Liu, A. Abraham and W. Zhang *A Fuzzy Adaptive Turbulent Particle Swarm Optimization*, International Journal of Innovative Computing and Applications p 39-47, 2007.

[44] C. Grosan, A. Abraham and M. Nicoara *Search Optimization Using Hybrid Particle Sub-Swarms and Evolutionary Algorithms*, International Journal of Simulation Systems, Science and Technology, UK, p 60-79, 2005.

[45] S. Das, A. Abraham and A. Konar *Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis*, Applications and Hybridization Perspectives, Advances of Computational Intelligence in Industrial Systems, Y. Liu et al. (Eds.), Studies in Computational Intelligence, Springer Verlag, Germany, p 1-38, 2008.

[46] D. H. Kim and A. Abraham *Hybrid Genetic Algorithm- Particle Swarm Approach for Function Optimization*, Hybrid Evolutionary Algorithms, C. Grosan et al. (Eds.), Studies in Computational Intelligence, Springer Verlag, Germany, p 147-170, 2007.

[47] A. Abraham, H. Guo and H. Liu *Swarm Intelligence: Foundations, Perspectives and Applications*, Swarm Intelligent Systems, N. Nedjah and L. Mourelle (Eds.), Studies in Computational Intelligence, Springer Verlag, Germany, p 3-25, 2006.

[48] Y. Maheshkumara, V. Ravi and A. Abraham *A Particle Swarm Optimization Threshold Accepting Hybrid Algorithm for Unconstrained Optimization*, Neural Network World, Volume 22, 2013.

[49] H. Liu, A. Abraham, V. Snasel and S. McCloone *Particle Swarm Scheduling for Work-Flow Applications in Distributed Data-Intensive Computing Environments*, Information Sciences, Elsevier Science, Netherlands, p 228-243, 2012.

[50] C. Grosan and A. Abraham *Hybrid Evolutionary Algorithms: Methodologies, Architectures and Reviews, Hybrid Evolutionary Algorithms*, Studies in Computational Intelligence, C. Grosan et al. (Eds.), Springer Verlag, Germany, p1-17, 2007.