

# Fuzzy Ant Supervised by PSO and Simplified Ant Supervised PSO Applied to TSP

Nizar Rokbani<sup>1,2</sup>, Aijth Abraham<sup>2,3</sup>, Adel.M Alimi<sup>1</sup>

<sup>1</sup>REGIM-Lab, REGIM: REsearch Groups on Intelligent Machines

University of Sfax, National Engineering School of Sfax (ENIS)

BP 1173, Sfax, 3038, Tunisia

<sup>2</sup>MIR Labs: Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, WA, USA

<sup>3</sup>IT4innovations, VŠB-Technical University of Ostrava, Czech Republic

{nizar.rokbani, aijth.abraham, adel.alimi}@ieee.org

**Abstract**—Bio-inspired techniques and swarm intelligence are used to solve complex problems. In this paper, two new variants of AS-PSO (Ant Supervised by Particle Swarm optimization) meta-heuristic are proposed and applied to a classical travelling salesman benchmark problem. The new variants are Fuzzy-AS-PSO and Simplified AS-PSO (S-AS-PSO). AS-PSO is a hierarchical meta-heuristic based on the ant colony optimisation (ACO) and particle swarm optimization (PSO), in which ACO is the heuristic and PSO is the meta-heuristic. The paper reviews the initial formulation; and introduces a new focus as well as two new variants. AS-PSO is an adaptive heuristic, since the user is not asked to fit any parameter values. In AS-PSO, the ACO algorithm is in charge of the problem solving, while the PSO is managing the optimality of the ACO parameters. The Simplified AS-PSO, S-AS-PSO, is a variant that uses simplified PSO while in Fuzzy AS-PSO; the fuzzy PSO is used as a meta-heuristic. The paper also includes an application of the new AS-PSO variants to the travelling Salesman Problem (TSP) and is compared with the ACO results.

**Keywords**— Fuzzy AS-PSO, Simplified AS-PSO, ACO, PSO, Meta Heuristics, TSP.

## I. INTRODUCTION

To find an optimal solution, several approaches could be investigated, including exact algorithms that work in order to have an optimal solution, with the disadvantage of reaching slowly to the solution, and heuristics that provide a feasible solution quickly, but not necessarily optimal. It is in this context that ant colony, ACO, and PSO are involved.

Searching for an optimum, is a challenging task starting by defining what makes a solutions optimal; optimality is quantified using mathematical and numerical criteria that should be fitted with respect to constraints, These criteria are expressed as a set of mathematical expressions admitting a minimum value, these expressions also called objective functions, fitness functions or coast functions. The heuristic optimizer, have, then, to propose a solution satisfying that function [1].

Within optimization methods we distinguish the heuristic algorithms; PSO, particle swarm optimization [2], and ACO, Ant Colony Optimization [3], are heuristics inspired from natural ants and animal social organisation.

Form this point of view, a problem could be solved by different techniques including heuristics and several analyses are needed to consider the best solution among them. Even within a specific class of heuristics, the optimality of a solution depends on the heuristic parameters [4].

On the other hand, finding a set of good parameters of an heuristic method is always a challenging problem, theoretical investigations as well as statistical experimentations are need to confirm a set a parameters for a specific class of problems. Adaptive variants tend to find the best set of heuristic parameters, while running, in order to limit their impact on the quality of the solutions.

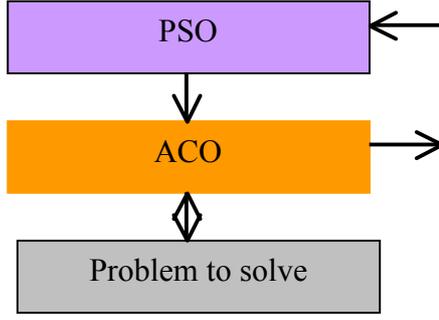
According to Talbi [5], the hierarchical heuristics could be seen as: low level hybrids and high level hybrids. In low-level hybridization, an internal function, or sub-processing, of an heuristic is replaced by another heuristic.

In high level hybridization, two heuristics are hybridized without affecting their respective internal functioning. Within these classes, two mechanisms could be involved:

- The Hierarchical mechanism, it is involved when heuristics are executed sequentially; the output of the first heuristic is used by the second one.
- Co-evolutionary mechanism, qualify an evolution in witch agents from different heuristics cooperate in parallel to explore the space of solutions. [5].

AS-PSO, is a High-level Hierarchical, HLH, meta-heuristic, since the output of the ACO heuristic is used by PSO to optimize ACO Parameters, see Figure 1 [6, 7].

The remaining of this paper is organized as follows: Section II gives a brief review on PSO and ACO algorithms. Section III, is dedicated to the presentation of AS-PSO, the initial proposition and new variants. Application of AS-PSO to the travelling salesman problem, around Tunisian cities, is presented in Section IV and the paper concludes with results, discussions and further work proposals.



**Figure 1.** AS-PSO Architectural Overview

## II. PSO AND ACO

### A. Particle swarm Optimization

PSO, is an optimization method inspired from the ability of social organisations to face and solve complex a problems involving the individual capacities of their members and also the group social dynamics. A group of individuals, which have a little intelligence, each one, can create a complex social intelligence using a set of simple interaction rules. The social organisation is called a swarm and the individual is called a particle [8].

In PSO, a particle is informed about its neighbourhood and also about the group. The best neighbourhood solution as well as the best global solution is used by the particle to adjust its forthcoming displacement. The particle displacement obeys to following equations [2].

$$\vec{V}_{i+1} = w\vec{V}_i + c_1 * rand() * (\vec{x}_{lbest} - \vec{x}_i) + c_2 * rand() * (\vec{x}_{Gbest} - \vec{x}_i) \quad (1)$$

$$\vec{x}_{i+1} = \vec{x}_i + \vec{V}_{i+1} \quad (2)$$

Equation (1) is composed of three terms, in the first one  $w$  is the inertia weight, It moderates the current velocity of the particle. The second term, is the cognitive component of displacement,  $(c1)$  moderates the cognitive behaviour of the particle. The last term represents the social behaviour and it is moderated by the factor  $(c2)$ ; It defines a displacement relative to the best solution of the group.

The PSO algorithm needs only a few parameters prior to run : the moderation factors values ( $w$ ,  $c1$  and  $c2$ ), the number of particles, the maximum number of iterations as well as a fitness function. The fitness function is used to evaluate the performance of the particles at each iteration. It is also used to elect the best neighbour and the best group solution.

### B. Ant Colony Optimization.

The ACO is inspired from the food search process of natural ants; It focuses on the use of phenomenon as a marker of the prospective paths. A good path leads to a food position, and will be marked several times by all Ants attending the food. The pheromone is here a link and a social interaction medium between the ants of a colony [3].

---

### Algorithm 1 Ant colony

---

ACO ( $K, N$ : Integer)

1 Initialize the positions of totally  $K$  ants, as well as the pheromone matrix  $\tau^{(0)}$

2 For the construction-step index  $n = 1 : N$

• For the ant index  $k = 1 : K$

\*Consecutively move the  $k$ -th ant for  $L$  steps, according to a probabilistic matrix  $p^{(n)}$  (with a size of  $M_1 M_2 \times M_1 M_2$ )

• update the pheromone matrix  $\tau^{(n)}$

3 Make the solution decision according to the final pheromon matrix  $\tau^{(N)}$

---

**Figure 2.** ANT algorithm.

Assume that ( $K$  ants) are involved to find the optimal solution in a space  $\chi$  that consists of  $M1 \times M2$  nodes. the procedure of ACO can be summarized as in Fig 2. In ACO, the displacement of an ant depends on the transition matrix  $p(n)$ , and the pheromone state matrix  $\tau(n)$ . The nodes of the transition matrix are computed iteratively using equation (3)., they define the probabilistic action rules.

$$p_{i,j}^{(n)} = \frac{(\tau_{i,j}^{(n-1)})^\alpha (\eta_{i,j})^\beta}{\sum_{j \in \Omega_i} (\tau_{i,j}^{(n-1)})^\alpha (\eta_{i,j})^\beta} \quad (3)$$

Where  $\tau(i,j)$  represents the pheromone value of the arc linking the node  $(i)$  to the node  $(j)$ ;  $\Omega_i$  is the neighbourhood nodes for a given ant  $a(k)$ . The constants  $\alpha$  and  $\beta$  represent the moderation of pheromone information and heuristic information  $\eta(i,j)$ , allowing to go from node  $(i)$  to node  $(j)$ . The pheromone matrix is updated during the ACO procedure; this update is performed after the displacement of ants, according to equation (4), where  $\rho$  is the evaporation rate,  $Bt$ , the best tour .

$$\begin{aligned} \text{if } (i,j) \in B_t &\longrightarrow \tau_{i,j}^{(n-1)} = (1 - \rho)\tau_{i,j}^{(n-1)} + \rho\Delta_{i,j}^{(k)} \\ \text{else} &\longrightarrow \tau_{i,j}^{(n-1)} = \tau_{i,j}^{(n-1)} \end{aligned} \quad (4)$$

A second pheromone update is performed by the end of the ants displacements; the whole pheromone matrix is updated as in equation (5) Where  $(\rho)$  is the pheromone decay coefficient..

$$\tau_{i,j}^{(n)} = (1 - \rho) \cdot \tau_{i,j}^{(n-1)} + \rho \cdot \tau_{i,j}^{(0)} \quad (5)$$

The resumed ACO processing steps appears in Figure 2.

### III. AS-PSO, ANT SUPERVISED BY PSO.

The proposed method, AS-PSO, An ACO is represented by its parameters; a set of parameters is a PSO particle. The ACO is the heuristic and the PSO is the meta-heuristic. The heuristic is directly related to the physical problem and try to solve it, while the meta-heuristic adjusts the parameters of the heuristic. So the operation of the heuristic, ACO in our case, is either guided or supervised by the meta-heuristic, PSO. From this organization, several variants of AS-PSO are proposed in the following paragraphs.

Classically for a given problem, ACO parameters are set by the user in the initialisation and remain the same. By adding a PSO we aim to slightly vary these parameters in order to smooth their impact on the solution quality. The impact of the user settings is, hence, reduced. Indeed, the system is responsible for providing all the necessary parameters using PSO. Figure 3, shows key processing steps of the AS-PSO algorithm.

#### A. The early AS-PSO proposal

AS-PSO global architecture was proposed [6], as a global meta-heuristic, with a local search policy assumed by ant colonies and a supervision of solutions assumed by PSO. ACO Swarms reported their results to a PSO who was in charge of finding the best solution as illustrated in Figure 4.

This architecture was built to fit the requirements of complex robotic systems, needing discrete actions, while evolving in a continuous space; it was adapted to a biped robot. Joint actuators are assimilated to intelligent agents, part of a complex hierarchical swarm. To optimise the joints displacement, an ACO is used. A hierarchical communication flowchart was also proposed; it allowed the reporting of all local solutions in order to generate a global synchronous compartment for a forward stable walk. This architecture was one among several proposals using PSO to solve the problem of gait generation in biped robotics [8, 9, 10, 11, 12].

#### B. The AS-PSO adjusting $[\alpha, \beta]$

A second proposal made in [7], more close to a numerical optimization model, by Rokbani et al. consists in adjusting the  $[\alpha, \beta]$  ACO parameters using a PSO. The PSO helps the ACO to find its best parameters and the PSO particle stands as:

$$X = [\alpha, \beta] \quad (6)$$

Classically in ACO, the constants  $\alpha$  and  $\beta$  represent the impact of pheromone information and heuristic information. In AS-PSO they are no more constant but subject to an optimization process in witch PSO is used. This made AS-PSO a kind of adaptive ACO.

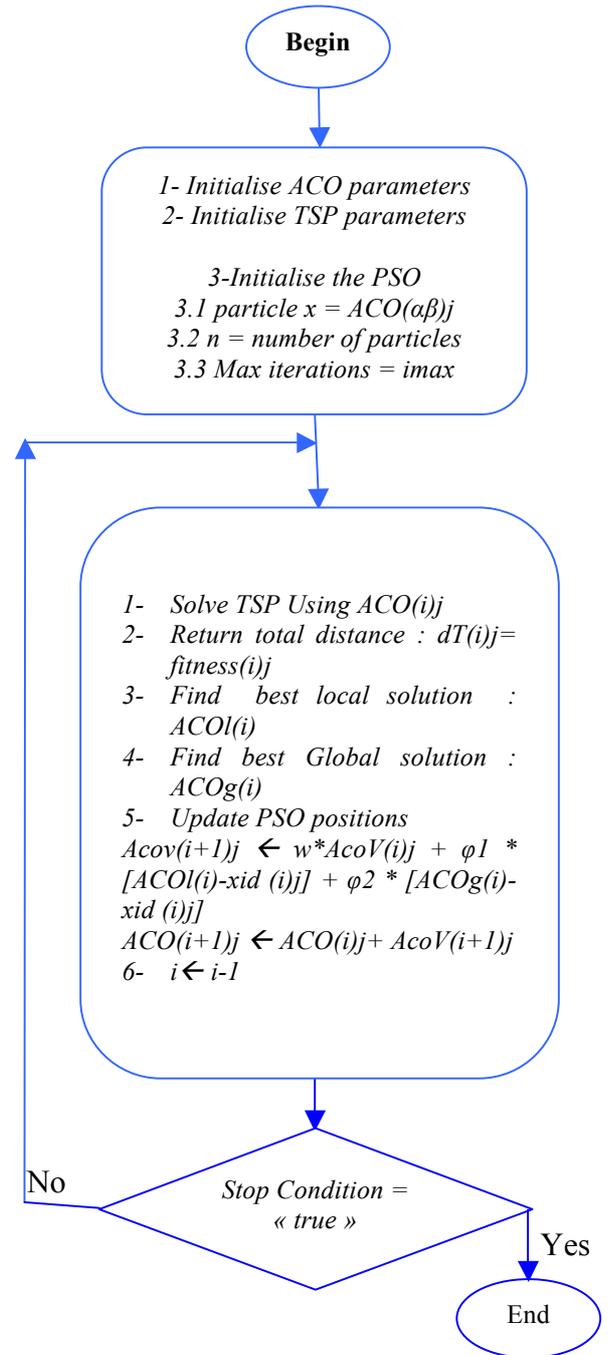


Figure 3. AS-PSO Algorithm overview

#### C. Simplified AS-PSO

Simplified AS-PSO variant, S-AS-PSO, are using the same nominal schema as the classical AS-PSO, while here the simplified PSO variant is used instead of the standard one. The

ACO remains similar to what was described in the previous paragraph.

Simplified PSO variants PSO-VG is proposed by Pedersen [13], It is a variant in which the particles velocity depends on the best global solution only, this means that cognitive contribution is completely ignored and the velocity equation is transformed into equation (7).

$$v_i = w*v_i + c_2*rand()*(p_{gbest} - x_i) \quad (7)$$

This version is called socially only PSO, this simplified PSO could reduce computational time and convergence for problem with a single global optimum. The cognitive aspect of the displacement is completely ignored, only the global best is used to moderate the individual comporment, this is a typical case of PSO interpretation [10].

A very close proposal to PSO-VG, is the PSO-G, where particles positions are updated directly using the global solution only, see equation (8). Here the velocity, is simply directly plugged into the position vector, once more the local best, representing the cognitive comporment is reduced.

$$x_{i+1} = x_i + c*rand()*(p_{gbest} - x_i) \quad (8)$$

Such a variants could be viewed as a PSO with a cognitive parameter ( $c1 = 0$ ), these variants are faster than the classical PSO since there is reduction in the algorithm, the processing needed to evaluate the best local is simply omitted.

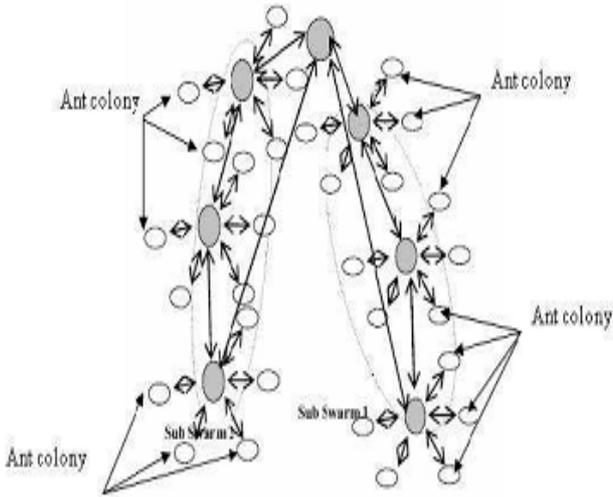


Figure 4. AS-PSO Architectural for robotic application [6]

#### D. Fuzzy AS-PSO

Fuzzy PSO is used to optimise the ANT parameters. In FPSO, the displacement toward the best global is moderated with a charisma factor. The global best is no more qualified as a single particle but as a set of neighbourhoods, each neighbourhood has its best solution and also receives a charisma factor that indicates its weight in the velocity update.

$$\vec{V}_{i+1} = w\vec{V}_i + c_1*rand()*(\vec{x}_{lbest} - \vec{x}_i) + \sum_{i=1}^K \psi_i*(c_2*rand()*(\vec{x}_{Gi} - \vec{x}_i)) \quad (9)$$

If the number of number hoods is limited to (1) and  $\psi=1$ , we can easily obtain the original form of a standard PSO, this is way the fuzzy PSO is seen as a generalization of PSO [14].

#### IV. APPLICATION TO TSP PROBLEM

##### A. The TSP Problem.

The travelling salesman problem, TSP, could be described as follows: assume a travelling salesman to which is assigned (n) cities that he should visit, strating from his home city and coming back to it by the end of its tour. The challenge consists in the design of a road map allowing to reduce the global distance, and to limit multiple visits to each city; optimally the TSP should visit a city just once [13].

To evaluate the potential strength of AS-PSO, the algorithm variants were applied to the TSP problem visiting Tunisian cities. Cities were appended from a map representation; a city is represented by its coordinates on the map, the distance is computed as Euclidian One, a typical solution using AS-PSO appears in figure 5.

##### B. Comparative results

Results presented in Table1, represent the average distance obtained when the algorithm is applied to 10, 20, 30 and 60 cities with the same test configuration using matlab software.

The experimentations concern classical AS-PSO, Simplified- AS-PSO (S-ASPSO) and Fuzzy-AS-PSO (F-AS-PSO), results are compared to the classical ACO algorithm, which serves as reference.

Table 1, shows that the results obtained by ACO and AS-PSO for ten cities are close to each other. The number of cities used here is very limited, and the solution is trivial compared to what is expected from an intelligent system. For 20 cities, the best result is obtained with S-AS-PSO, with a very close fitness to F-AS-PSO. All AS-PSO variants clearly overcome the results of ACO. For 30cities the best variant is the S-AS-PSO, followed by the Fuzzy variant, the classical AS-PSO and finally the ACO. For 60 cities, the Fuzzy- AS-PSO gives clearly the best result for the shortest path; S-AS-PSO takes the second position, AS-PSO the third and finally the ACO.



Figure 5. AS-PSO applied to Tunisian cities.

TABLE I. COMPARITIVE RESULTS FOR 10, 20, 30 AND 60 CITIES.

Number of cities	Fitenss (total distance)			
	ACO	AS-PSO(I)	S-AS-PSO	F- AS-PSO
10	3569.22	3063.45	3120.32	3154.68
20	4061.78	3259.37	3240.36	3244.72
30	5588.61	4310.98	4606.54	4298.35
60	10469.97	8446.11	8223.94	8207.95

## V. CONCLUSIONS

The ACO showed its capacity for solving the TSP problem with a fixed set of parameters. With AS-PSO, we tried to adjust automatically the parameters of ACO using a PSO. In this paper some new AS-PSO variants are proposed including simplified- AS-PSO, (S-AS-PSO) and Fuzzy AS-PSO (F-AS-PSO) with comparative results.

Preliminary results showed that in all cases AS-PSO variants give better results than classical ACO. These results need to be confirmed using a large statistical analysis.

For the TSP problem, when the number of cities becomes larger (60), the Fuzzy AS-PSO gives better results than ACO and the other AS-PSO variants. This hypothesis should be

confirmed using larger statistical tests. From the preliminary experimentations, conducted in this work, the simplified AS-PSO performed with mid-range problem scale, here 20 to 30 cities. This hypothesis also needs a larger test bench to validate the results.

## ACKNOWLEDGMENTS

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. Ajith Abraham acknowledges the support from the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 funded by Structural Funds of the European Union and state budget of the Czech Republic.

## REFERENCES

- [1] N. Rokbani and A. M. Alimi. "IK-PSO, PSO Inverse Kinematics Solver with Application to Biped Gait Generation.International", Journal of Computer Applications, vol 58, number (22), pp: 33-39, November 2012.
- [2] Y. Shi, and R. Eberhart. "A modified particle swarm optimizer". In Proc of the 1998 IEEE World Congress on Computational Intelligence and IEEE International Conference on Evolutionary Computation, pp: 69-73, 1998.
- [3] M. Dorigo, M. Birattari, and T. Stutzle. "Ant colony optimization.", IEEE Computational Intelligence Magazine (2006), 28-39, 2006.
- [4] M. Reimann, and M. Laumanns. "A hybrid aco algorithm for the capacitated minimum spanning tree problem." Proceedings of first international workshop on hybrid metaheuristics. 2004.
- [5] Talbi, E.-G. Taxonomy of hybrid metaheuristics. Journal of heuristics, vol. 8, no 5, pp: 541-564, 2002.
- [6] W. Elloumi, N. Rokbani and A.M. Alimi, " Ant supervised by PSO", . In Proc of International symposium on Computational Intelligence and Intelligent Informatics, pp: 161-166, 2009.
- [7] N. Rokbani, A. L. Momasso, and A. M. Alimi, "AS-PSO, Ant Supervised by PSO Meta-heuristic with Application to TSP". Proceedings Engineering & Technology-Vol, 4, pp: 148-152, 2013.
- [8] N.Rokbani, E. Benbousaada, B. Ammar, B., & A. M. Alimi. "Biped robot control using particle swarm optimization". In IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 506-512, 2010.
- [9] N. Rokbani, A. M. Alimi, and B. Ammar. "Architectural Proposal for a Robotized humanoid, IZiman". In IEEE International Conference on Automation and Logistics, pp : 1941-1946, 2007.
- [10] N. Rokbani, and A. M. Alimi. "Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis". Procedia Engineering, Elsevier, 2013.
- [11] M. Aghaabbasloo, M. Azarkaman and M.E. Salehi,. "Biped robot joint trajectory generation using PSO evolutionary algorithm". In AI & Robotics and 5th RoboCup Iran Open International Symposium (RIOS), pp : 1-6, 2013.
- [12] N. Rokbani, Boussada, E. B., B.A. Cherif, and A. M. Alimi. "From gaits to ROBOT, A Hybrid methodology for A biped Walker". In Proc of Clawar 2009, Vol. 12, pp. 685-692, 2009.
- [13] M.E.H Pedersen and A.J Chipperfield, "Simplifying particle swarm optimization". In Applied Soft Computing, volume 10, Issue 2, March 2010, pp. 618-628, Elsevier.
- [14] Abdelbar, Ashraf M., Suzan Abdelshahid, and Donald C. Wunsch. "Fuzzy PSO: a generalization of particle swarm optimization." Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on. Vol. 2. IEEE, 2005.