International Journal of Hybrid Intelligent Systems 15 (2019) 17–26
DOI 10.3233/HIS-180258
IOS Press

Solving the travelling salesman problem using fuzzy and simplified variants of ant supervised by PSO with local search policy, FAS-PSO-LS, SAS-PSO-LS

Nizar Rokbani[a,b,c,*], Ajith Abraham[c], Ikram Twir[b] and Abdelkrim Haqiq[d]
[a]*REGIM-Lab, National Engineering School of Sfax, University of Sfax, Sfax, Tunisia*
[b]*ISSAT Sousse, University of Sousse, Sousse, Tunisia*
[c]*Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, Auburn, WA 98071, USA*
[d]*FST, Hassan 1st University, Settat, Morocco*

Abstract. Combinatorial optimization problems have several industrial applications such as Network routing, IOT network routing, path Planning for robotics and manufacturing for which the travelling Salesman Problem, TSP, can serve as typical test bench. This paper investigates new variants of the Fuzzy Ant Supervised by PSO, FAS-PSO and Simplified Ant Supervised by PSO, SAS-PSO coupled with a local search, Ls, mechanism. The proposed method is based on the Fuzzy PSO to supervise and tune ACO parameters, in addition to a local search mechanism helping in avoiding cities local crossing. The SAS-PSO-Ls uses the same idea while with the simplified PSO as supervisor. Experimentations (a space is missed before "Experimentations") and results are based TSP test benches with a statistical analysis and a comparative study with the standard AS-PSO and similar state of art methods. FAS-PSO-Ls gives better than the state of art for eil51, berlin52, while the SAS-PSO-Ls is giving better results for the following cases: eil51, berlin52, st70.

Keywords: TSP, Heuristics, hybridization, combinatorial problems, IOT, Fuzzy PSO, ACO

1. Introduction

Several industrial applications such as Network routing, IOT network routing, Robotics, Path planning [24,31] are typical combinatorial optimization problems for which the Traveling Salesman Problem, TSP, could serve as a test bench.

TSP, is a popular optimization problem in which a salesman has to visit a set of predefined cities and return to his first position without crossing the same city more than a time. Multiple Traveling Salesman Problem, MTSP, was proposed in [32] as a solution to solve default detection in an IOT network. The MTSP consists in dividing the problem into many sub-problems each one assumed to TSP, then controlling each subproblem individually.

Bio-inspired techniques showed their capacities in solving such a problem such as particle Swarm Optimization [16], Ant Colony Optimization [15], Firefly algorithm [12] and flower pollination algorithm [38].

Bi-heuristic approaches consist in the use of two techniques and collaboration strategy, among them the high-level hybridization or a low-level hybridization. The low-level hybridization consists in modifying an internal function of a heuristic by another heuristic. For the high-level hybridization, the meta-heuristic is

*Corresponding author: Nizar Rokbani, REGIM-Lab, National Engineering School of Sfax, University of Sfax, Sfax, Tunisia. E-mail: Nizar.rokbani@ieee.org.

running to tune the second heuristic parameters while second heuristic is used to solve the problem [5]. Ant Colony Optimization was combined with the genetic algorithm in [8], the genetic algorithm is used to propose a better solution for the next generation and to help him to avoid trapping in a local optimum. The water flow, WF, and the Tabu Search, TS, was combined to find the best global solution for the traveling salesman problem, where the WF is used to enhance the search space exploration and the TS is used to ameliorate the search space exploitation [39]. In [4], A multiple ants' clan was proposed in hybridization with the genetic algorithm, ACOMAC, to solve the traveling salesman problem, for large TSP test benches, authors proposed to add two concepts which are the multiple nearest neighborhoods, and the dual nearest neighborhoods, DNN, to ameliorate the search process. In [29], a modified real-valued antibody network, RABNET, was proposed to solve the TSP, the algorithm is a hybridization between a neural network algorithm and immune system algorithm, RABNET-TSP is an unsupervised neural network with a single layer using learning clone selection and affinity mutation. In [33], the original RABNET-TSP was modified to solve large size TSP test benches, the main contribution consists in enhancing the original variant by using a threshold to activate the core function and then using a winner' selection mechanism. The new version [33] ameliorates the previous one [29], in term of time consuming. A hybridization between ACO and the delete cross method was proposed in [35], to solve TSP, the cross delete method is added to increase the convergence speed of ACO. The fuzzy hybrid heuristic methods such in [34] are based on the hybridization of a heuristic by another fuzzy one. In [25], authors used the fuzzy PSO variants of [26] to self-adapt ACO parameters, as well as a simplified AS-PSO version was proposed to solve the TSP Problem based on the Tunisian map. In [22], AS-PSO version was proposed to solve the standard TSP test benches. In [3], authors proposed a close variant based on the same hybridization architecture called Ant supervised by PSO with a local search policy 3opt, in which the PSO was used to optimize ACO settings while ACO is used to solve the TSP with the support of 3-Opt as a local search algorithm allowing avoiding trapped in local optima. In [6], authors proposed the fuzzy ant colony optimization to solve the traveling salesman problem, the algorithm consists on using the Fuzzy logic system, FLS, to modify the evaporation coefficient "ρ", mainly FLS was used to control the algorithm search process [6]. In [28], authors combined the FPSO with the simulated annealing, SA, to solve the TSP. Authors used the fuzzy system and the SA to avoid the possibility to get trapped in local optimum. A self-adaptive ant system was developed in [20,36] to solve optimization problems. In [30], Kefi et al. investigated the PSO inertia weight to parametrize ACO settings. ACO is running to find the global best tour of TSP, where a local search algorithm was inserted to improve the global best and to ignore the local optimum. In [10], Authors aim was to solve the identification of plants pieces problem based on Particle Swarm Optimization and leaf biometrics. In [2], A hybridization schema of bat algorithm and direct search methods were used to solve to solve minmax problems. This paper conducts investigations on a new bi-heuristic technique which belongs to the high-level hybridization class called Ant supervised by Fuzzy PSO-Ls, FAS-PSO-Ls, in order to solve the Traveling salesman problem. The proposal is compared to the Simplified Ant supervised by PSO with local search, SAS-PSO-Ls and related techniques.

The remaining of this paper is organized as follows: Section 2 is reserved to the problem statement. Section 3 is detailing the techniques and methods used in this paper are briefly presented, this includes the ant colony optimization, ACO, variant, the PSO fuzzy variant, as well as the local search mechanism which is used to enhance the algorithm efficiency. Section 4 details the used architecture as well as the developed algorithm. In Section 5, comparative experimentations are presented and commented, the paper ends by conclusions and perspectives.

2. Problem statement

The traveling Salesman Problem, TSP, is a famous combinatorial optimization problem, in which we have to reach all cities once and return to the start point with minimum cost and distance. Many researchers run for the TSP applying their algorithms to prove its convergence or its solutions quality thanks to the availability of its common test benches, see Eq. (1). PSO, ACO, and close bio-inspired algorithms are largely used to solve the TSP test benches [11].

$$F = \sum_{i=1,j=2}^{N} \sum_{i \neq j}^{N} x_{ij} c_{ij} \tag{1}$$

$$x_{ij} = \left\{ \begin{matrix} 1 \textit{ if the path goes from city } i \textit{ to city } j \\ 0 \textit{ otherwise} \end{matrix} \right\} \tag{2}$$

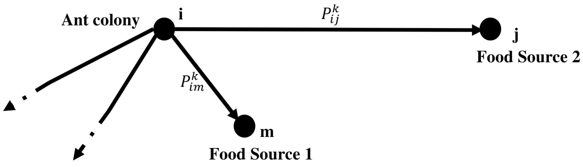


Fig. 1. Ant strategy.

$$c_{ij} = ||x_i - x_j|| \tag{3}$$

F is the total distance of a tour, see Eq. (1), N presents the number of the cities list, x_{ij} is a binary coefficient, equal to 1 if an arc linking city i to city j is existent and equal to 0 otherwise, see Eq. (2). The distance between city i and city j is expressed using Eq. (3), which is an euclidian distance. Equations (4) and (5) prove that each city is visited once.

$$\sum_{j=1, i \neq j}^{N} x_{ij} = 1, i = 1, 2, 3, \ldots, N \tag{4}$$

$$\sum_{i=1, i \neq j}^{N} x_{ij} = 1, j = 1, 2, 3, \ldots, N \tag{5}$$

3. Methods and techniques

3.1. Ant Colony Optimization ACO

Ant Colony Optimization, ACO, was inspired from the natural ants' capacities in searching a food source, as well as an optimum path joining the food source to ant's colony [18]. Ants use a biological marker called pheromone to collaborate and communicate with each other's in order to find the shortest tour to their food. Naturally, ants choose the path with the highest pheromone quantity to pass for the next node. ACO individuals use a probability equation to pass from the current position to the next one, see Eq. (6). The node which is connected to an arc with the highest probability, will be chosen [18,36]. To update the pheromone, ants use Eq. (7), Fig. 1.

$$P_{i,j}^k = \frac{(\tau_{i,j}^{k-1})^\alpha * \eta_{i,j}^\beta}{\sum_{j \in \Omega i} (\tau_{i,j}^{k-1})^\alpha * \eta_{i,j}^\beta} \tag{6}$$

$$\begin{aligned} &\textit{if } (i,j) \in \textit{ Best Tour } \tau_{ij} = (1-\rho)\, \tau_{ij}^{(k-1)} \\ &\quad +\rho \Delta_{ij}^k \;\; \textit{else } \tau_{ij}^{(k-1)} = \tau_{ij}^{(k-1)} \end{aligned} \tag{7}$$

W here $\tau i, j$ denotes the pheromone quantity between two nodes j, and i, Ω_i represents the i_{th} neighborhood, α, β are the substrate parameters, $P_{i,j}^k$ stands for the probability of k_{th} ant passing the path (i, j), where ρ stands the Pheromone decay coefficient.

3.2. Fuzzy and simplified Particle Swarm Optimization

Eberhart and Kennedy proposed the Particle Swarm Optimization, PSO in 1995, which is based on bird flocks and fish banks behavior [14,27]. In PSO, a swarm is made of particles with a communication media allowing to get informed about the best solution of a particle neighbors and the best swarm solution. PSO particles move using the following equations, see Eqs (8) and (9).

$$\begin{aligned} v_i &= wv_i + C1 * \textit{rand}() * (P_{\textit{lbest}} + x_i) \\ &\quad + C2 * \textit{rand}() * (P_{\textit{gbest}} + x_i) \end{aligned} \tag{8}$$

$$x_i = x_i + v_i \tag{9}$$

Where w is the inertia weight of PSO, C1, and C2 stands respectively for the cognitive coefficient and the social coefficient, $P_{\textit{lbest}}$, $P_{\textit{gbest}}$ are the best local and global positions. x_i stands for the position of particle i and v_i stands for the velocity of particle i.

3.2.1. Simplified Particle Swarm Optimization, SPSO

Simplified Particle Swarm Optimization is a PSO variant in which the local solution obtained by the neighborhoods is ignored while the best solution obtained from the global community is considered, means removing completely the cognitive coefficient and considering only the social coefficient. PSO individual positions and velocities are ruled using Eqs (10) and (11). Pedersen introduced the Simplified PSO variants, PSO-VG in [19].

$$v_i = wv_i + C2 * \textit{rand}() * (P_{\textit{gbest}} + x_i) \tag{10}$$

$$x_i = x_i + v_i \tag{11}$$

3.2.2. Fuzzy Particle Swarm Optimization, FPSO

The Fuzzy Particle Swarm Optimization, is a PSO variant, based on a Fuzzy Logic inference system incorporated into the classical PSO, enhancing the search space exploration and exploitation. In this paper, we are using the generalized Fuzzy PSO algorithm proposed in [3]. FPSO is running to find the global best tour based on a charismatic factor. Means, the best solution is evaluated by k neighborhoods. Each neighborhood finds a best solution and receives a charismatic factor. FPSO particles update their positions and velocities using the following equations, Eqs (12)–(14).

$$v_i = wv_i + C1 * rand\,() * (P_{lbest} + x_i) + \sum_{i=1}^{k} \Psi_i (C2 * rand() * (P_{gbest} + x_i)) \quad (12)$$

$$\Psi_i = \frac{1}{1 + \frac{x_i - P_{gbest}}{\beta_{PSO}}} \quad (13)$$

$$\beta_{PSO} = \frac{f\,(P_{gbest})}{l} \quad (14)$$

Where k is random selected individuals from PSO swarm, Ψ_i presents charismatic factor for an individual i, l isa fixed parameter by the user, x_i stands for $(\alpha, \beta, \rho)^T$.

Many FPSO versions were proposed in the literature such as [1,37]. The Fuzzy PSO, FPSO is incorporated in the flowchart to tune ACO parameters, While ACO runs to solve the traveling Salesman Problem. When the heuristic gives a better tour for the TSP with tuned parameters, the meta-heuristic runs to find the best ACO optimized parameters using the fitness function.

3.3. Local search

In [7], Croes proposed K-Options algorithm, KOpt, allowing to select the shortest among k options by removing k connections from a node and reconnect them to the next node with respect to the tour construction. 2-Opt is a particular K-opt case in which we remove two connections and reconnect them in other way. The algorithm accepts the new Tour only if it is shorter than the previous best tour. 2-Opt local search policy helps the heuristic to avoid local optimum and runs to find the global one, see Fig. 2.

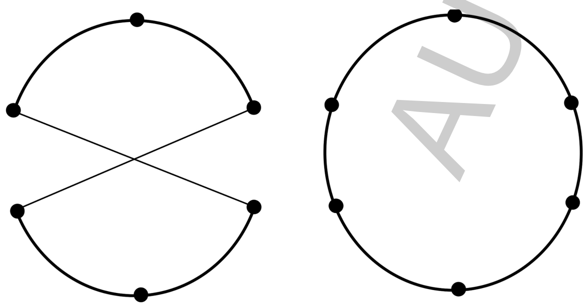


Fig. 2. Fuzzy ant supervised by PSO Local search.

4. FAS-PSO-Ls

The fuzzy ant supervised by PSO, FAS-PSO, was introduced by Rokbani et al. [25]. Globally, PSO is running to optimize ACO parameters while ACO is running to find the optimal solution for the main problem, here the target is to optimize the global tour passed by a salesman, the standard TSP test benches are used for comparisons needs. FAS-PSO-Ls is detailed using Fig. 3.

Table 1
Fuzzy PSO initial Parameters

	Fuzzy Particle Swarm Optimization
Population	10
Maximum iterations	100
Parameters	$C1 = 1.5$
	$C2 = 1.5$
	$W = 0.6$
FPSO particle	$(\alpha, \beta, \rho)^T$

FPSO particle is equal to the triple ACO parameters $(\alpha, \beta, \rho)^T$, FPSO particles number is equal to the number of ACO instances. The algorithm is running with a fixed number of iterations (in our case = 100). After initializing FPSO, we have to initialize ACO parameters where the number of ants is equal to the number of city, as well as FPSO, ACO is running with a fixed number of iterations.

FPSO algorithm is aiming to search for the best ACO solutions and running to find the best ACO parameters. The results of ACO parameters using FPSO are used to tune ACO, search, find and evaluate the obtained solutions, where a local search is integrated into ACO helping him to ameliorate its solutions. FAS-PSO-Ls continue its execution until satisfied criteria, which is the maximum iterations number is reached or the global tour found by the proposal is less than the Best-Known Solution, BKS, for each TSP test bench. FAS-PSO-Ls is detailed with Fig. 3.

5. Experimental investigations

5.1. Experimental protocol

For experimentationsa personal computer PC, Intel Core^{TM} 2, 4 GB RAM size, is used to runMATLAB Software, R2015a, to evaluate the algorithm, FAS-PSO-Ls, efficiency in term of solution quality and time execution we have to calculate the mean, μ, the standard deviation, SD, see Eq. (15) and the error, err., for TSP selected test benches according to the best-known solution, BKS. The error, err., is ruled using Eq. (14) as in [26,30]. The selected test benches form the standard library TSPLib [9] are eil51, berlin52, st70, eil76, rat99, eil101, kroA100. Statistical analysis is done over

Algorithm 1: Ant Supervised by fuzzy PSO Algorithm with local search mechanism

1) Initialize FPSO parameters
2) Generate initial population of PSO P_k= (0,……., xn)
3) Gbest=Input (): Initialize Global Best
4) **For** i=1 **to** n **do** (n particles)
5) Initialize positions
6) Evaluate positions using ACO algorithm with local Search(2-opt)
7) Save initial values of ACO parameters
8) Update Global Best
9) **End For**
10) **For** t=0 **to** maximum iteration **do**
11) **For** i=1 **to** n **do (all n particles)**
12) Update Velocity using equation (12)
13) Update positionusing equation (11)
14) Evaluate positions using ACO algorithm with local Search(2-opt)
15) Update Local Best Lbest
16) Update Global Best Gbest
17) **End For**
18) **If** (Lbest<GBest) then
19) GBest=Lbest
20) **End For**
21) **Return** GBest, if maximum iteration is reached or GBestis equal or less than KBS

Fig. 3. Fuzzy ant supervised by PSO Local search.

Table 2
Optimized ACO parameters for each test bench (FAS-PSO-Ls)

Problems	α	β	ρ
eil51	0.5	1	0.01
berlin52	0.5	1	0.01
st70	0.5	1	0.01
eil76	0.5	1	0.01
rat99	0.5	1	0.01
eil101	0.5	1	0.01
kroA100	0.5	1	0.01

Table 3
Optimized ACO parameters for each test bench (SAS-PSO-Ls) [39]

Problems	α	β	ρ
eil51	0.9775	5	0.21663
berlin52	0.5	3.2138	0.27447
st70	1.9836	3.479	0.026371
eil76	1.5119	4.2564	0.41424
rat99	–	–	–
eil101	1.7963	4.7321	0.22427
kroA100	1.6549	3.3073	0.32603

10000 iterations using the TSP test benches for comparative results.

With ant swarm size equal to city number, the fuzzy PSO is running with social and cognitive coefficients respectively C2, and C1 fixed both to 1.5, and the inertia weight equals to 0.6. FPSO swarm size is equal to 10 with a maximum iteration equal to 100. For a statistical analysis, we are using the MATLAB software statistics toolbox [17]. Results are presented under statistical analysis with error establishing as in Eq. (15) and standard deviation as in Eq. (16).

$$Err = ((\mu - BKS) \div BKS) * 100 \tag{15}$$

$$SD = \sqrt{\left(\frac{1}{T}\sum_{t=1}^{T}(x_t - \mu)^2\right)} \tag{16}$$

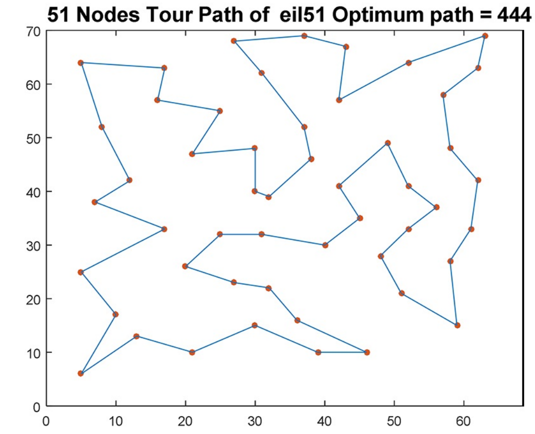
Where SD stands for the standard deviation, T stands for the number of iterations, x_t presents the best solution in iteration, t, and μ presents the mean.

FAS-PSO-Ls algorithm is running with a set of parameters mentioned in Table 1.

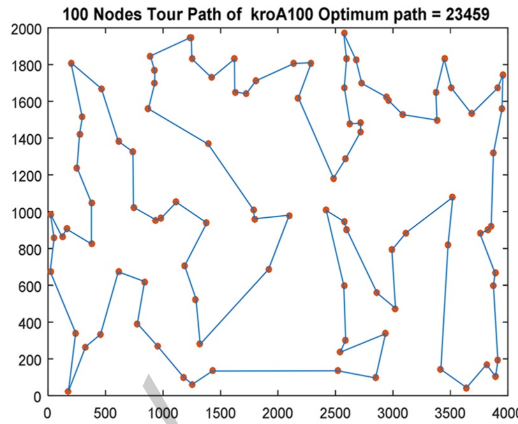
5.2. Experimental results

With mentioned Fuzzy PSO set of parameters, FPSO optimized ACO settings using his fitness function, see Eq. (9). ACO parameters for each test bench are listed with Table 2. ACO optimized parameters are the set of parameters with which ACO must give the shortest global tour. Best α, β and ρ given by FPSO are respectively equal to 0.5, 1, and 0.01 for all TSP test benches.

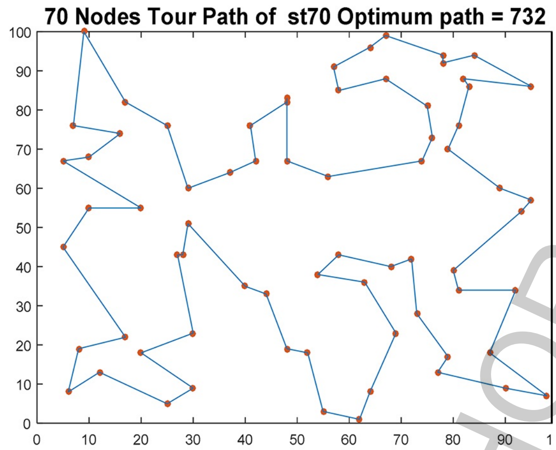
To show the global best tour obtained by FAS-PSO-Ls foreach test bench, we are using a "Draw City" function; in which the red points represented the cities in a cartesian frame and the blue lines represented the path passed between two correspondent cities. Table 4, line 13 details the best Tour for each TSP test



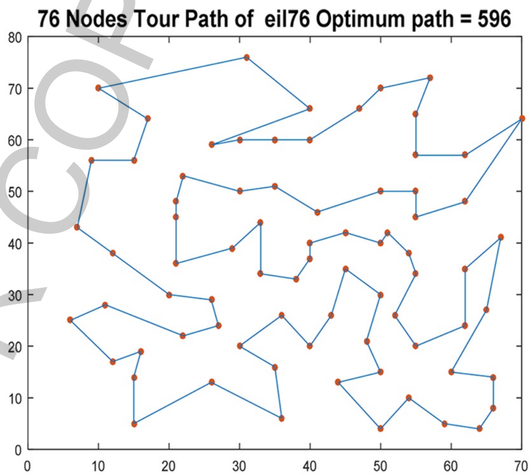
a. Eil51 global best tour using FAS-PSO-2Opt



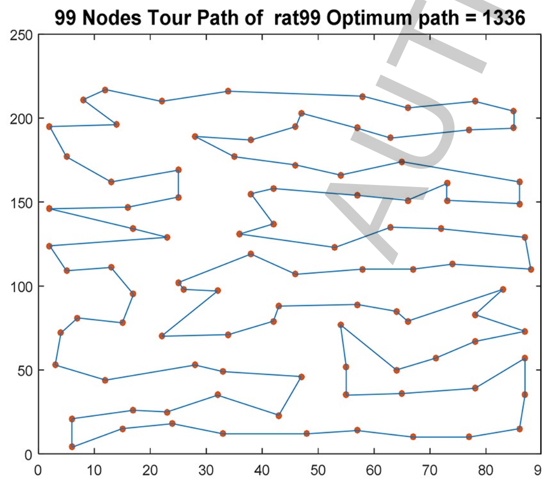
b. KroA100 global best tour using FAS-PSO-2Opt



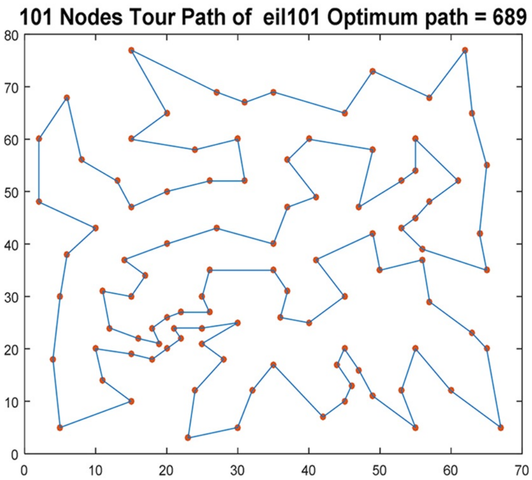
c. St70 global best tour using FAS-PSO-2Opt



d. Eil76 global best tour using FAS-PSO-2Opt



e. Rat99 global best tour using FAS-PSO-2Opt



f. Eil101 global best tour using FAS-PSO-2Opt

Fig. 4. FAS-PSO-2Opt obtained global best tour.

Table 4
Fuzzy Ant Supervised by PSO-2OPT comparative results

	Problem	eil51	berlin52	st70	eil76	rat99	eil101	kroA100
	BKS	426	7542	675	538	1211	629	21282
ACOMAC	Avg.	430.68	–	–	555.70	–		21457.00
(2004) [11]	SD	–	–	–	–	–		–
	Error (%)	1.10	–	–	3.29	–		0.82
RABNET –	Avg.	438.70	8073.97	–	556.10	–	654.83	21868.47
TSP (2006) [12]	SD	3.52	270.14	–	8.03	–	6.57	245.76
	Error (%)	2.98	7.05	–	3.36	–	4.11	2.76
Modified RABNET –	Avg.	437.47	7932.50	–	556.33	–	648.63	21522.73
TSP (2009) [13]	SD	4.20	277.25	–	5.30	–	3.85	93.34
	Error (%)	2.69	5.18	–	3.41	–	3.12	1.13
VRS – 2Opt	Avg.	431.10	7547.23	–	–	–	648.67	21498.61
(2012) [23]	SD	–	–	–	–	–	–	–
	Error (%)	1.2	0.07	–	–	–	3.13	1.02
ACO – 2Opt	Avg.	439.25	7556.58	–	–	–	672.37	23441.80
(2012) [14]	SD	–	–	–	–	–	–	–
	Error (%)	3.11	0.19	–	–	–	6.90	10.15
Hybrid ACO	Avg.	431.20	7560.54	–	–	1241.33	–	–
(2012) [14]	SD	2.00	67.48	–	–	9.60	–	–
	Error (%)	1.22	0.23	–	–	2.5	–	–
GA-Ant System	Avg.	–	7634.00	–	542.00	–	–	21437.00
(2012) [9]	SD	–	–	–	–	–	–	–
	Error (%)	–	1.22	–	0.74	–	–	0.73
ACO-Tagushi	Avg.	435.40	7635.40	–	565.50	–	655.00	21567.10
	Method [37]	SD	–	–	–	–	–	–
	Error (%)	2.21	1.24	–	5.11	–	4.13	1.34
ACO-ABC	Avg.	443.39	7544.37	700.58	557.98	–	683.39	22435.31
(2015) [38]	SD	5.25	0.00	7.51	4.10	–	6.56	231.34
	Error (%)	4.08	0.03	3.79	3.71	–	8.65	5.42
PSO–ACO–3Opt	Avg.	426.45	7543.20	678.20	538.30	1227.40	632.70	21445.10
(α, β) (2015) [18]	SD	0.61	2.37	1.47	0.47	1.98	2.12	78.24
x	Error (%)	0.11	0.02	0.47	0.06	1.35	0.59	0.77
AS-PSO	Avg.	428	7542	678	541	1236	632	21457
2opt [24]	SD	9.97	202.62	15.92	12.16	31.74	12.29	391.85
	Error (%)	0.23	0.0	0.44	0.55	2.08	0.47	0.82
SAS-PSO-Ls [39]	Avg.	**426**	**7542**	**675**	543	–	645	**21305**
	SD	**9.6403**	**206.1429**	**20.6865**	13.1426	–	12.1358	**674,4597**
	Error (%)	**0**	**0**	**0**	0.92937	–	2.5437	**0,10807**
FAS-PSO-Ls	Avg.	**427**	**7542**	678	552	1259	653	21466
	SD	**10.5469**	**254.5967**	20.7752	12.2927	31.1385	12.2107	–
	Error (%)	**0.23474**	**0**	0.44444	2.6022	3.9637	3.8156	0.86458

benches and its error compared to the best-known solution, BKS.

The figures listed from Fig. 4a to f illustrated the best global tour obtained by the algorithm for selected TSP test benches, eil51 global best tour is equal to 444, see Fig. 4a. kroA100 optimized Tour is equal to 23459, see Fig. 4b. The best global path for st70 is equal to 732, which is represented by Fig. 4c. The best global tour for eil76 is equal to 596 which is illustrated with Fig. 4d. FA-PSO-Ls gives a global solution equal to 1336 for rat99, which is illustrated with Fig 4e. Eil101 best solution is equal to 689, see Fig. 4f.

In term of solution quality, FAS-PSO-Ls gives fair results for small TSP test benches such as eil51, berlin52 and st70, and gives acceptable results for TSP large size configurations such as eil76, rat99, eil101, as well as kroA100.

Compared to the simplified FAS-PSO-Ls, and in term of Time execution, FAS-PSO-Ls is more time consuming. For example, for berlin52 in which the best-known solution is obtained, the time consumed for 100 epocs is equal to 6985.174097 seconds which is equal to 2 hours, where the time execution to obtain the BKS for Simplified AS-PSO is 1883.653. For a comparative result, investigations are assuming that the

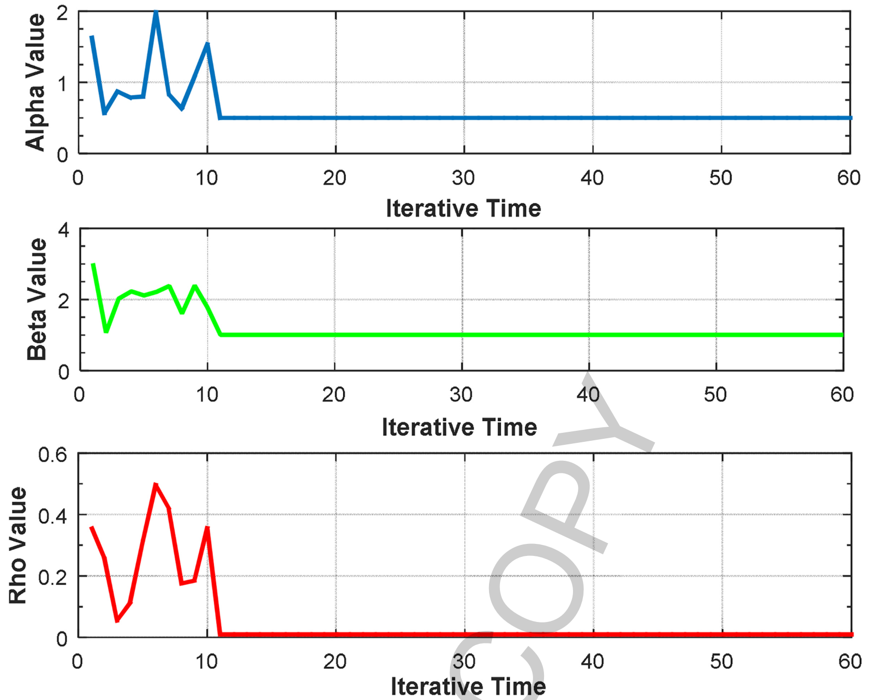


Fig. 5. ACO parameters evolution using Fuzzy PSO (berlin52, antnum = 52).

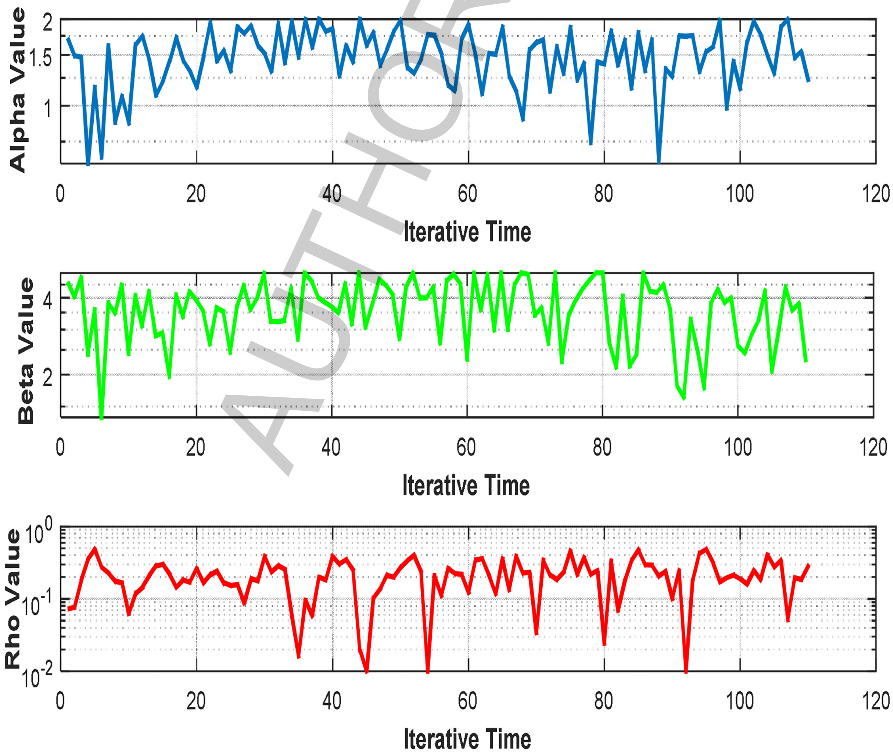


Fig. 6. ACO parameters evolution using simplified PSO (berlin52, antnum = 52).

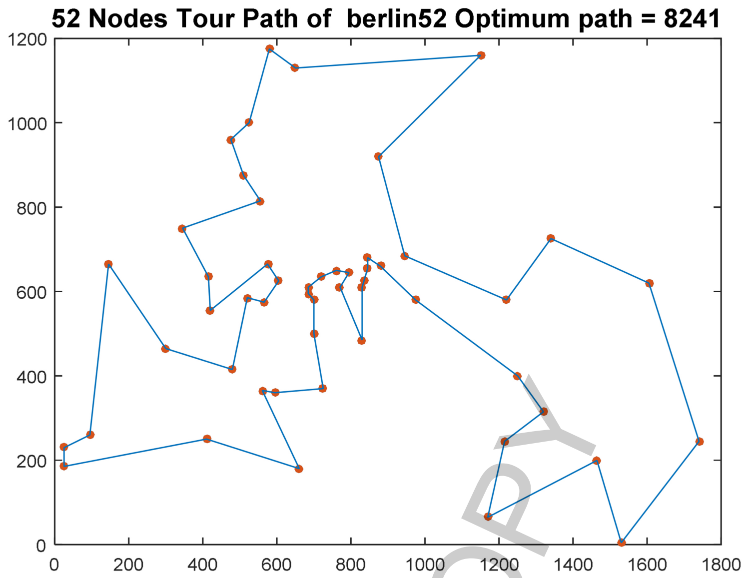


Fig. 7. Berlin52 global best tour using FAS-PSO-2Opt.

simplified Ant supervised by PSO with a local search mechanism is better than the FAS-PSO-Ls in term of solution quality as well as the time speed for the following cases: eil52, berlin52, st70, eil76, rat99, see Table 4 lines 12 and 13.

In term of ACO parameters convergence, Fig. 5 showed that the FAS-PSO-Ls is suffering from parameters stagnation, where ACO parameters variations are observed in the early iterative processing of the proposal and are then maintained in the remaining of the processing, a perturbation mechanism will be investigated to avoid such behavior. Figure 7 represents the best tour of berlin52.

The SAS-PSO- performs more explorations of ACO parameters as in can be seen in Fig. 6, Table 3. Compared to the state of art, see Table 4, FAS-PSO-Ls gives better than related techniques for the following cases: eil51, berlin52, and st70.

6. Conclusions and prespectives

Experimental investigations of the fuzzy ant supervised by particle swarm optimization with a local search mechanism, FAS-PSO-LS were exposed in this paper to solve the traveling salesman, problem, TSP. The proposal is compared to the ant supervised by PSO, AS-PSO, the simplified Ant Supervised by PSO, SAS-PSO-Ls and similar techniques. Compared to the state of art, FAS-PSO-Ls gives better results in the following cases: eil51, berlin52, and st70, see Table 4.

Further investigations will focus on solving IOT devices using FAS-PSO-Ls with the use of other local search policies such as the variable neighborhoods, as well as using a new architecture based on the fuzzy logic system.

References

[1] A. Alfi and M.M. Fateh, Intelligent identification and control using improved fuzzy particle swarm optimization, *Expert Systems with Applications* **38**(10) (2011), 12312–12317,

[2] A.F. Ali and M.A. Tawhid, Hybrid bat algorithm and direct search methods for solving minimax problems, *International Journal of Hybrid Intelligent Systems* (2018), 1–15.

[3] A.M, Ashraf, S. Abdelshahid and D.C. Wunsch, Fuzzy PSO: a generalization of particle swarm optimization, in: *Proceedings 2005 IEEE International Joint Conference on Neural Networks IJCNN'05,* Vol. 2, IEEE, 2005, pp. 1086–1091.

[4] C.F. Tsai, C.W. Tsai and C.C. Tseng, A new hybrid heuristic approach for solving large traveling salesman problem, *Information Sciences* **166**(1) (2004), 67–81.

[5] E.G. Talbi, A Taxonomy of Hybrid Metaheuristics, *Journal of Heuristics* **8** (2002), 541–564.

[6] F. Olivas, F. Valdez and O. Castillo, Ant Colony Optimization with Parameter Adaptation Using Fuzzy Logic for TSP Problems, in: P. Melin, O. Castillo and J. Kacprzyk, eds, Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization, *Studies in Computational Intelligence*, Vol 601, Springer, Cham, 2015, pp. 593–603.

[7] G.A. Croes, A method for solving traveling salesman problems, *Operations Res* **6** (1958), 791–812.

[8] G.F. Dong, W.W. Guo and K. Tickle, Solving the traveling salesman problem using cooperative genetic ant systems, *Expert System Application* **39** (2012), 5006–5011.

[9] G. Reinelt, TSPLIB – A Traveling Salesman Problem Library, *ORSA Journal on Computing* **3**(4) (1991), 376–384.

[10] H.F. Eid and A. Abraham, Plant species identification using leaf biometrics and swarm optimization: A hybrid PSO, GWO, SVM model, *International Journal of Hybrid Intelligent Systems* **14**(3) (2017), 155–165.

[11] İ. İlhan, An Application on Mobile Devices with Android and IOS Operating Systems Using Google Maps APIs for the Traveling Salesman Problem, *Applied Artificial Intelligence* **31**(4) (2017), 332–345.

[12] I. Twir, N. Rokbani and A. Alimi, Ant Supervised by Firefly Algorithm with a Local Search Mechanism, ASFA-2Opt, *2018 – IEEE International Conference on Control, Automation and Diagnosis*, 2018.

[13] I. Twir, N. Rokbani, A. Haqiq and A. Abraham, Experimental Investigation of Ant Supervised by Simplified PSO with Local Search Mechanism (SAS-PSO-2Opt), in: A. Abraham, A. Haqiq, A. Muda and N. Gandhi, eds, Proceedings of the Ninth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2017), SoCPaR 2017. Advances in Intelligent Systems and Computing, vol 737. Springer, Cham, 2018.

[14] J. Kennedy and R. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[15] K. Jun-Man and Z. Yi, Application of an improved Ant Colony Optimization on generalized Traveling Salesman Problem, *Energy Procedia* **17** (2012), 319–325.

[16] K.P. Wang, L. Huang, C.G. Zhou and W. Pang, Particle swarm optimization for traveling salesman problem, International, *Conference on Machine Learning and Cybernetics* **3** (2003), 1583–1585.

[17] MATLAB Statistics Toolbox User's Guide. 2014, The MathWorksInc http:www.mathworks.com/help/pdf_doc/stats/stats.pdf.

[18] M. Dorigo, M. Birattari et al., Swarm intelligence, *Scholarpedia* **2**(9) (2007), 1462,

[19] M.E.H. Pedersen and A.J. Chipperfield, Simplifying particle swarm optimization, *Applied Soft Computing* **10** (2010), 618–628.

[20] M. Förster, B. Bicke, B. Hardung and G. Kókai, Self-Adaptive Ant Colony Optimisation Applied to Function Allocation in Vehicle Networks, in: *Proceedings of the 9th annual conference on Genetic and evolutionary computation,* 2007, pp 1991–1998.

[21] M. Gunduz, M.S. Kiran and E. Ozceylan, A hierarchic approach based on swarm intelligence to solve traveling salesman problem, *Turkish Journal of Elec Eng & Computer Sciences* **23** (2015), 103–117.

[22] M. Mahia, Ö.K. Baykanb and H. Kodazb, A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem, *Applied Soft Computing* **30** (2015), 484–490.

[23] M. Peker, B. Sen and P.Y. Kumru, An efficient solving of the traveling salesman problem: The ant colony system having parameters optimized by the Taguchi method, *Turkish Journal of Elec Eng & Computer Sciences* **21** (2013), 2015–2036.

[24] M.V. Narayana, Route Optimization by using Multiple Travelling Sales Person Problem in MANETs, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* **3**(1) (2018), 782–790.

[25] N. Rokbani, A. Abraham and A.M. Alimi, Fuzzy Ant Supervised by PSO and Simplified Ant Supervised PSO Applied to TSP, *The 13th International Conference on Hybrid Intelligent Systems, HIS 2013*, Gammarth, Tunisia, December 4–6, IEEE 2013, ISBN 978-1- 4799-2438-7, 2013.

[26] N. Rokbani, A.L. Momasso and A.M. Alimi, AS-PSO, Ant Supervised by PSO Meta-heuristic with Application to TSP, *Proceedings Engineering & Technology* **4** (2013), 148–152.

[27] N. Rokbani and A.M. Alimi, Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis, *Procedia Engineering* **64**(Supplement C) (2013), 1602–1611.

[28] R.F. Abdel-Kader, Fuzzy Particle Swarm Optimization with Simulated Annealing and Neighborhood Information Communication for Solving TSP, *International Journal of Advanced Computer Science and Applications (IJACSA)* **2**(5) (2011), 593–603.

[29] R. Pasti and L.N. De Castro, A Neuro-Immune Network for Solving the Traveling Salesman Problem, *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006, pp. 3760–3766.

[30] S. Kefi, N. Rokbani, P. Krömer and A.M. Alimi, Ant supervised by PSO and 2-opt algorithm, AS-PSO-2Opt, applied to traveling salesman problem, in: *IEEE International conference on System Man and Cybernetics SMC (2016)*, IEEE, 2016, pp. 004866–004871.

[31] S. Trigui, O. Cheikhrouhou, A. Koubaa, U. Baroudi and H. Youssef, FL-MTSP: a fuzzy logic approach to solve the multi-objective multiple traveling salesman problem for multi-robot systems, *Soft Computing*, Springer, **21**(24) (2017), 7351–7362.

[32] S. Zhou, K.J. Lin and C.S. Shih, Device clustering for fault monitoring in Internet of Things systems, *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 228–233.

[33] T.A.S. Masutti and L.N. de Castro, A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem, *Inf Sci* **179**(10) (2009), 1454–1468.

[34] W. Elloumi, N. Baklouti, A. Abraham and A.M. Alimi, Hybridization of fuzzy PSO and fuzzy ACO applied to TSP, in: *13th International Conference on Hybrid Intelligent Systems (HIS*), December, 2013, pp. 106–111.

[35] W. Junqiang and O. Aijia, A Hybrid Algorithm of ACO and Delete-Cross Method for TSP, *The Proceedings of the 2012 International Conference on Industrial Control and Electronics Engineering*, 2012, pp. 1694–1696.

[36] W. Ying and X. Jianying, An Adaptive Ant Colony Optimization Algorithm and Simulation [J], *Acta Simulata Systematica Sinica* **1** (2002), 9.

[37] Y. Shi and R.C. Eberhart, Fuzzy adaptive particle swarm optimization, in: *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, 2001, pp. 101–106.

[38] Y. Zhou, R. Wang, C. Zhao, Q. Luo and M.A. Metwally, Discrete greedy flower pollination algorithm for spherical traveling salesman problem, *Neural Computing and Applications*, Springer, 2017, pp. 1–16.

[39] Z.A. Othman, A.I. Srour, A.R. Hamdan and P.Y. Ling, Performance water flow-like algorithm for TSP by improving its local search, *International Journal of Advancements in Computing Technology* **5** (2013), 126–137.