International Journal of System Dynamics Applications
Volume 9 • Issue 3 • July-September 2020

A Hybrid Hierarchical Heuristic-ACO With Local Search Applied to Travelling Salesman Problem, AS-FA-Ls

Nizar Rokbani, High Institute of Applied Science and Technology of Sousse, University of Sousse, Tunisia & REGIM-Lab, University of Sfax, Tunisia & National Engineering School of Sfax, Tunisia

Pavel Kromer, Department of Computer Science, FEECS, VSB - Technical University of Ostrava, Czech Republic

Ikram Twir, High Institute of Applied Science and Technology of Sousse, University of Sousse, Tunisia

Adel M. Alimi, REGIM-Lab, University of Sfax, Tunisia & National Engineering School of Sfax, Tunisia

ABSTRACT

The combinatorial optimization problem is attracting research because they have a wide variety of applications ranging from route planning and supply chain optimization to industrial scheduling and the IoT. Solving such problems using heuristics and bio-inspired techniques is an alternative to exact solutions offering acceptable solutions at fair computational costs. In this article, a new hierarchical hybrid method is proposed as a hybridization of Ant Colony Optimization (ACO), Firefly Algorithm (FA), and local search (AS-FA-Ls). The proposed methods are compared to similar techniques on the traveling salesman problem, (TSP). ACO is used in a hierarchical collaboration schema together with FA which is used to adapt ACO parameters. A local search strategy is used which is the 2 option method to avoid suboptimal solutions. A comparative review and experimental investigations are conducted using the TSP benchmarks. The results showed that AS-FA-Ls returned better results than the listed works in the following cases: berlin52, st70, eil76, rat99, kroA100, and kroA200. Computational investigations allowed determining a set of recommended parameters to be used with ACO for the TSP instances of the study.

KEYWORDS

ACO, Ant Supervised By FA ASFA, ANT Supervised By Firefly With Local Search, AS-FA-Ls, FA, Firefly Algorithm, Local Search, Travelling: Salesman Problem

INTRODUCTION

The advent of self-driving cars, intelligent transportation systems, and internet of things devices routing challenges, recalled the interest in combinatorial optimization problems, COP. Among them, the traveling salesman problem is still attracting interest since it can stand for many industrial applications such as internet of things networks routing, components placements on board for electronics manufacturing, transportation systems (Bajracharya,2016), containers management & logistic optimization chains, robotics (Rajasekaran et al., 2014), etc. Bio-inspired techniques such as Flower Pollination algorithm by (Yang, 2009), Particle Swarm Optimization by (Kennedy & Eberhart, 1995), Ant Colony Optimization by (Dorigo & Birattari, 2007), Firefly Algorithm by (Yan, 2010) and Article Bee Colony algorithm by (Karaboga, 2005) showed their capacities to solve such problems.

DOI: 10.4018/IJSDA.2020070104

Hybrid heuristics are in general built by combining several heuristics algorithms in order to tackle the weaknesses of each one and to improve the quality of the solutions essentially in engineering design problems (Moussa et al., 2015; Pradhan, 2017). Hybridization is already based on a collaboration schema between heuristics. Hierarchical heuristics can be seen as: low level hybrids and high-level hybrids. In low-level hybridization, an internal function, or sub-processing, of a heuristic is replaced by another heuristic. In high level hybridization, two heuristics are hybridized without affecting their respective internal functionality. The hierarchical mechanism is involved when heuristics are executed sequentially; the output of the first heuristic is used by the second one, while the co-evolutionary mechanism leads to an evolution in which agents from different heuristics cooperate to explore the space of solutions in parallel (Rokbani et al., 2013), and where authors proposed an early hybridization of ACO and PSO, to overcome the sensibility of ACO to parameters.

In (Wahid et al., 2018), the authors proposed a combination between the FA, the genetic algorithm, GA, and the Pattern Search, PS, with application to standard mathematical functions. In this approach GA was used to generate a set of solutions that were later evolved for a fixed number of iterations by the firefly algorithm. The evolved solutions were evaluated and if an optimum solution was not found, the GA was recalled to modify the population on the basis of its classical operators, crossover, and mutation. The pattern search was introduced to moderate the solution obtained by the FA. The proposed hybridization was tested using the standard test functions such as the Ackley, Rosenbrock, and the Sphere functions.

A hybridization schema based on the Ant Colony System, ACS, and the Firefly algorithm, FA, was proposed in (Goel & Maini, 2017) with application to Vehicle Routing Problem, VRP. A Hybrid Ant Firefly Algorithm, HAFA, consisting in using ACS to generate the initial problem solutions' representing the pool for FA was used to improve the search space exploration ability. The best solution found by the FA was used to update the pheromone trail. Here, the authors proposed a discrete version of FA to adapt it to the discrete problem. A combination of ACO and FA has been proposed in (Olief et al., 2016) in which ACO looked for the global solution and the FA focused on local optima using its neighborhood mechanism for TSP. The FA was the first method to search for local best tours and then the ACO was involved to search for the best tour on top of the FA results. In (Ariyantne et al., 2016), the authors used the firefly algorithm to optimize ACO settings. They aimed to find the global best path by optimizing α, β, and ρ; while not using any local search mechanism. In (Kumbharana & Pandey, 2013) the authors proposed the use of FA to solve the TSP. The fireflies were in this approach encoded in a discrete finite space representation, as vectors of (n) positions. When a firefly was supposed to move based on the distance to its neighbor, the Hamming distance was used to look after the best configuration among a given set of cities. The mechanism efficiently allowed avoiding minima. The method was applied to several TSP problems limited to 51 cities, including the city10, 16, 23, 30, and Eli51 TSP instances.

Making ACO self-adaptive using heuristics or bio-inspired techniques was proposed by (Rokbani et al., 2013) where particle swarm optimization was used to adapt the ACO parameters dynamically. The hybrid strategy was named ANT Supervised by PSO, since PSO is assumed to a supervisor of the ACO heuristic. Ant supervised by PSO with a local search mechanism including 3Opt was proposed in (Mahi et al., 2015) AS-PSO-2Opt while in (Kefi et al., 2016) the same architecture is used with the 2 Opt local search policy, AS-PSO-2OPT. The classical AS-PSO was applied to the standard TSP test instances. Several variants of PSO where investigated including fuzzy PSO and simplified PSO such in (Rokbani et al., 2019a). The gravitational PSO with a local search mechanism was also investigated in (Rokbani et al., 2019b).

In this study, a new hierarchical hybrid algorithm based on the combination of ACO with FA and local search, AS-FA-Ls, is designed. The proposed methods are compared to similar techniques using TSP test problems.

The remainder of this paper is organized as follows: Section 2 details the problem to be solved, the TSP. Section 3 reviews the key aspects of ACO and presents the FA as well as the local search heuristic. In Section 4, the architectural scheme of the proposed methods is presented. Section 5 is reserved to the experimental investigations and comparative results while conclusions and further works appear in Section 6.

The Traveling Salesman Problem

Combinatorial optimization problems from a class of problems consisting in solving discrete optimization tasks by minimizing or maximizing a cost Function with respect to specific constraints. Such problems are widely found in industrial and robotics applications. The TSP is an optimization problem in which a salesman aims to visit a set of cities and return to his initial city. He is required to pass each city exactly once and the cost of the travel, which can be related to the total distance of the tour, is supposed to be minimized. Several meta-heuristics algorithms have been applied to solve TSP. TSP Fitness Y consists in finding the best global tour passing all cities and return to the start point such as in (Ilhan, 2017) (Laporte et al., 1990), see Equation (1).

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

$$x_{ij} = \begin{Bmatrix} 1 & if\, the\, path\, goes\, from\, city\, i\, to\, city\, j \\ 0 & otherwise \end{Bmatrix} \tag{2}$$

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

Where *xt* stands for the total tour distance (tour length), N represents the number of cities, x_{ij} is a binary coefficient; equal to 1 if an arc is going from city i to city j and equal to 0 otherwise, c_{ij} is the distance between city i and city j, c_{ij} stands for the Euclidian distance, Equations (4) and (5) specify the constraint that each node (city) is visited only once.

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

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

Note that X_i is the planar city position $X_i = \left(x_i, y_i\right)$ on a map.

Ant Colony Optimization

Ant Colony Optimization is a discrete heuristic inspired from natural ants organisation (Dorigo & Gambardella, 1997; Dorigo & Birattari, 2007). Ants are assumed to move in a search space composed of a grid of nodes and arcs, such as the one showed in Figure 1. The ants deposit a pheromone amount when they pass an arc joining a pair of consecutive positions. The ants move from an initial position, the location of the nest, towards a source of food and then move back to their starting position. This makes the ACO heuristic suitable for solving the TSP. At the end of the optimization, the trail with the maximum deposit of pheromone is assumed to be the best path joining a pair of locations. The ants are supposed to move from a node to another based on a probabilistic rule, where $P_{i,j}^{k}$ is the probability that an ant, k, will be passing the arc (i, j) joining node (i) to node (j), $\tau_{i,j}$ represents the amount of pheromone between nodes i and j. $\eta_{i,j}$ is the desirability of edge i, j. Ω_{i} denotes i_{th} neighbourhood, α is a coefficient to control the impact of τ_{ij}. β is a coefficient to control the impact of $\eta_{i,j}$.

$$P_{i,j}^{k} = \frac{(\tau_{i,j}^{k-1})^{\alpha} * \eta_{i,j}^{\beta}}{\sum_{j\epsilon©i} (\tau_{i,j}^{k-1})^{\alpha} * \eta_{i,j}^{\beta}} \quad (6)$$

The ant select its movement from a city, i, according to the probability are $P_{i,o}^{k}$, $P_{i,j}^{k}$, $P_{i,l}^{k}$, and $P_{i,m}^{k}$, as illustrated in Figure 1.

The pheromone acts as a marker helping in search space exploration. It provides a global map of the most visited arcs within a discrete search space. Such, arcs are associated with the optimum trajectory. The iterative pheromone deposition process is managed by Equation (7) where ρ is the pheromone decay coefficient such as in (Dorigo & Gambardella, 1997).

$$if\left(i,j\right) \in Best\ \ Tour\ \tau_{ij} = \left(1-\rho\right)\tau_{ij}^{(k-1)} + \rho"_{ij}^{k}\ else\ \tau_{ij}^{(k-1)} = \tau_{ij}^{(k-1)} \quad (7)$$

Firefly Algorithm

Firefly is a swarm intelligence algorithm inspired by the firefly's behaviour. Yang proposed this algorithm in (Yang, 2010). In nature, fireflies flash the light using bioluminescence processes essentially for reproduction needs. The females use this mechanism to attract males while in the artificial Firefly Algorithm, this process is simply assumed to a communication medium. The FA expects all individuals to be unisex with attractiveness is decreasing with distance. The light intensity, I, is in FA defined by Equation (8).

$$I = I_{0} e^{-\gamma r} \quad (8)$$

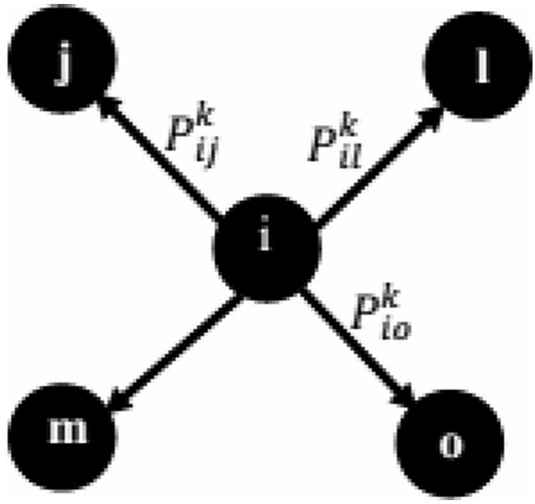
Where γ is the light absorption coefficient, I_0 is the initial intensity, and r is the distance between two fireflies.

The attractiveness of a firefly (i) is calculated according to Equation (9) where $^{2}{}_{0}$ is the initial attractiveness and γ is the light absorption coefficient:

$$\beta_{i} = \beta_{0} e^{-\gamma r^{2}} \quad (9)$$

The distance r between two fireflies i and j is given by Equation (10):

Figure 1. Probabilistic engagement Ant Strategy



$$r_{ij} = x_i - x_j = \sqrt{\sum_{k=1}^{N}\left(x_{i,k} - x_{j,k}\right)^2} \tag{10}$$

Firefly positions are updated according to Equation (11), where α is the mutation coefficient and λ is a random value:

$$x_i = x_i + \beta * rand(\ )\left(x_i - x_j\right) + \alpha * \lambda \tag{11}$$

The FA processing starts by defining fitness function with respect to the solve problem (Rokbani et al., 2015), as well as the brightness of the firefly and the maximum number of iterations and the way the random initial population is generated. The optimization process starts with a random population of fireflies. Then, the positions of the individuals are iteratively updated according to Equation (11). The process ends if a stopping criterion or the maximum number of iterations number is achieved. A simplified pseudo code of the FA is given in Figure 2.

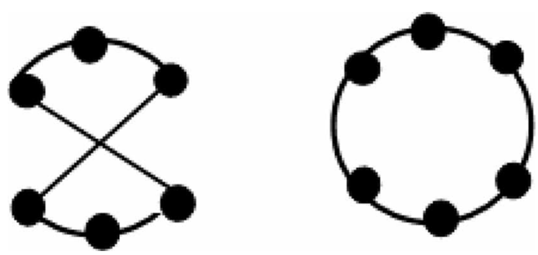
The 2 Options Local Search Policy (2Opt)

The K option (K-Opt) is a local search algorithm which removes for each node in a graph K connections and reconnects them in other positions (Croes, 1958). The most important feature of K-Opt, is that it is an iterative method which can be inserted within a heuristic without modifying its structure (Helsgaun, 2009). A popular K-opt variant is the 2Opt such as in (Dorigo & Stutzle, 2004). It consists in removing 2 arcs (connections), which means that 4 nodes are free, and will be reconnected; see Figure 3 (left). The nodes are then reconnected and the obtained path is retained only if it decreases the global distance used to interconnect the nodes like in Figure 3 (right). The 2 Opt variant is interesting since the processing involved is limited to four nodes, making the possible connections equal to four (4), among them two (2) are removed because of integrity violation; the nodes are not fully connected. The remaining two possibilities are simple to process since one is already the starting configuration. This explains how the proposal is effective in terms of time and computing.

Figure 2. Simplified FA pseudo code

```
Begin
Generate an initial population of fireflies  ;.
Formulate light intensity I so that it is associated with
      Define absorption coefficient γ
   While (t < MaxGeneration)
     for i = 1 : n (all n fireflies)
       for j = 1 : i (n fireflies)
         if (Ii> Ij),
           Vary attractiveness  ;
           move firefly i towards j;
           Evaluate new solutions and update light intensity;
         end if
       end for j
     end for i
     Rank fireflies and find the current best;
  end while
  Return (Best Firefly)
End
```

Figure 3. Local Search policy (2Opt)



The Hybrid AS-FA-Ls Proposal

ACO is a discrete heuristic so naturally adapted to solve combinatorial and discrete optimization problem, while ACO parameters are continuous in nature. This motivates the use of a continuous heuristic to fit the ACO parameters. The proposed methods are based on a couple of heuristic, ACO, and a meta-heuristic, FA. The heuristic acts as a solver while the meta-heuristic is in charge of parameters fitting (α, β and ρ). The flowchart in Figure 4, details the proposed approach.

The FA particle is set to ACO parameters $(\alpha, \beta, \rho)^T$, The FA swarm size is set equal to the number of ACO instances, the maximum number of FA iterations is set to a fixed value (for example, 100), the initial light absorption coefficient is fixed to one, the attractiveness, and the mutation coefficient are initially set to random values. The ACO parameters are initialized and the ACO population size is set equal to the number of cities. The maximum number of ACO iterations is fixed; FA fitness function is set to the TSP tour distance which is to be minimized such as in Equation (1).

EXPERIMENTAL INVESTIGATIONS

Experimental Protocol

The experimental protocol is based on a statistical analysis of the performances of the hybrid methods proposed in this study on the TSP problem. It is then compared to related works. A Matlab implementation and simulations are used to evaluate the performances in term of solution quality and convergence rate. The experiments were conducted using Matlab software, R2013a, running on a personal computer with i7-8709 CORE INTEL processor with 3.10 GHz processor speed and 8GB RAM size. For any given TSP instance, the route length over N cities is computed using Equation (1), for comparisons the best tour length as well as the average tour length (avg) and the standard deviation (SD) are used, (Hunt et al., 2014).

The following TSPLib as in (Reinelt, 1991) test TSP instances were used: eil51, berlin52, st70, eil76, rat99, kroA100, eil101, ch150, kroA200.

The standard deviation, SD, is calculated using the Matlab predefined function "std", where T is the number of iterations, x_t is the best position in each iteration t, μ is the average solution (avg), see Equations (12) and (13). Both (Avg) and (SD) are computed over 100 tests.

$$SD = \sqrt{\left(\frac{1}{T}\sum_{t=1}^{T}\left(x_t - \mu\right)^2\right)} \tag{12}$$

$$Avg = \mu = \frac{\sum_{t=1}^{T} x_t}{T} \tag{13}$$

To compare the proposal to similar techniques the result error is computed; it stands for the difference between the averages found solution, Avg, to the best-known solution, BKS (Reinelt, 1991) as in Equation (14).

$$Error = \left(\left(Avg - \ BKS\right) \div BKS\right)*100 \tag{14}$$

The heuristics parameters used for the simulations are listed in Table 1.

RESULTS AND DISCUSSIONS

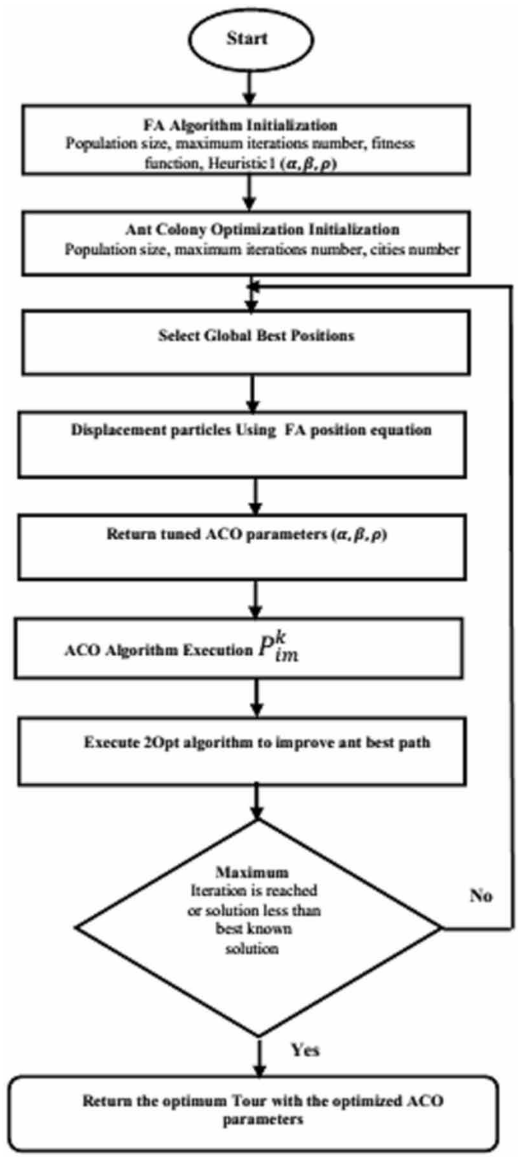
Optimum Tour Length

Compared to the state of art results, AS-FA-Ls gives better results than the listed solvers for the following cases: berlin52, st70, eil76, rat99, kroA100, and kroA200, see Table 2, last line for comparative results.

For example, using the FA as in (Kumbharana & Pandey, 2013) and (Ariyantne et al., 2016), The AS-FA-Ls, give a better result for berlin 52 and Kroa 100: for the remaining of the test instances AS-FA-Ls returned fair solutions for all test benches used in this study especially for berlin52, st70 and kroA100, see Table 3 for detailed results.

Figure 5 (a) shows berlin52 best tour which is obtained at a fitness of 7794; the optimum tour for the st70 problem is 683, shown in Figure 5 (b). For the eil76 test problem an optimum solution

Figure 4. ACO-FA -Local Search (2Opt) flowchart



is observed with a tour length of 549, see Figure 5 (c). The best Tour of rat99, illustrated by Figure 5 (d), is equal to 1258.

A couple of Kroa test instances were also investigated. In particular, Kroa 100 and Kroa 200, with tour lengths of 21821 and 29532, respectively. The method achieved better results than reference methods listed in Table 3 but did not reach the best-known solutions. Kroa 100 and 200 best tours are respectively visible on Figures 5(e) and 5(d), respectively.

Ability of Self-Tuning ACO Parameters

The berlin52 TSP instance, shown in Figure 6, was selected to investigate the ability of FA to tune ACO parameters. The instance was selected on the basis of good results with an error of about (0.46),

Table 1. Operated test conditions for AS-FA-Ls

	Firefly Algorithm
Iterations	100
Population	10
Parameters	$\alpha_0 = 2 * rand$ $\beta_0 = 0.2 * rand$ $\gamma = 1$

Table 2. Comparisons of the proposed algorithm with other techniques (BKS: Best Known Solution)

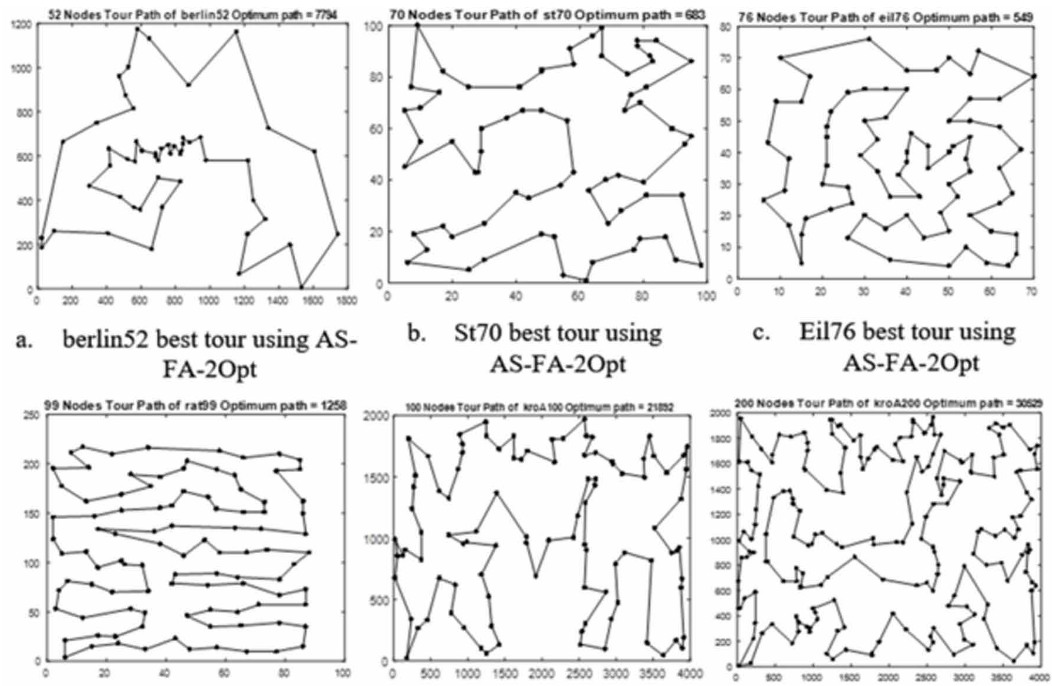
	Problem BKS	**eil51426**	**berlin527542**	**st70 675**	**eil76538**	**rat991211**	**eil101629**	**kroA10021282**	**ch1506528**	**kroA20029368**
ACO-ABC (Gunduz et al., 2015)	Avg.	443.39	7544.37	700.58	557.98	-	683.39	22435.31	6677.12	-
	SD	5.25	0.00	7.51	4.10	-	6.56	231.34	19.30	-
	Error (%)	4.08	0.03	3.79	3.71	-	8.65	5.42	2.28	-
PSO–ACO–3Opt (Mahia et al., 2015)	Avg.	426.45	7543.20	678.20	538.30	1227.40	632.70	21445.10	6563.95	29646.05
	SD	0.61	2.37	1.47	0.47	1.98	2.12	78.24	27.58	114.71
	Error (%)	0.11	0.02	0.47	0.06	1.35	0.59	0.77	0.55	0.95
AS-PSO 2opt (Kefi et al., 2016)	Avg.	428	7542	678	541	1236	632	21457	6560	29837
	SD	9.97	202.62	15.92	12.16	31.74	12.29	391.85	171.90	359.28
	Error (%)	0.23	0.0	0.44	0.55	2.08	0.47	0.82	0.49	1.60
ACSFA (Ariyantne et al., 2016)	Avg.	432.6	-	-	-	-	-	21390.71	-	-
	SD	-	-	-	-	-	-	-	-	-
	Error (%)	-	-	-	-	-	-	-	-	-
SAS-PSO-Ls (Twir and Rokbani, 2017)	Avg.	426	7542	675	543	-	645	21305	6606	29924
	SD	9.6403	206.1429	20.6865	13.1426	-	12.1358	674,4597	176.5837	631.5875
	Error (%)	0	0	0	0.92937	-	2.5437	0,10807	1.1949	1.8932
FA-TSP (Kumbharana & Pandey, 2013)	Avg.	435.60	-	-	-	-	-	-	-	-
	SD	-	-	-	-	-	-	-	-	-
	Error (%)	-	-	-	-	-	-	-	-	-
AS-FA-LS	**Avg.**	**428**	**7542**	**675**	**541**	**1213**	**639**	**21282**	**6571**	**29532**
	SD	**8.3575**	**-**	**15.1956**	**11.1445**	**29.1416**	**12.1791**	**437.8311**	**159.1596**	**628.9033**
	Error (%)	**0.46948**	**0**	**0**	**0.55762**	**0.16515**	**1.5898**	**0**	**0.6587**	**0.55843**

achieved by AS-FA-Ls. The dynamic evolution of ACO parameters by the FA is illustrated in Figure 7. The figure illustrates that the FA has introduced a great diversity of values and their combinations to ACO parameters.

The Analysis showed that the ACO (ρ) parameter is stabilized at the value of (0.7) at the level of iteration (60), see Figure 7 (c). At the same time high search diversity is observed for α, and β parameters as it can be observed in Figures 7 (a) and 7 (b). The same behaviour was observed also for other TSP instances used in this research.

A part of the study consisted also in gathering the set of best parameters that the proposed optimization methods used for ACO. There is no set of generic parameters suitable for all test TSP instances. Different sets of best performing parameters can be considered for each one of them. The recommended parameters for investigated TSP instances are listed in Table 4. The investigations showed that a redundant α parameter is (0.7) which is observed in 5/9 times while the value of 7 is returned in the remaining cases, $\beta \epsilon \left(0,5,9\right)$ while $\rho = \left(0.07, 0.03, 0.0963\right)$. This justifies the

Figure 5. Best tour obtained using AS-FA-Ls



a. berlin52 best tour using AS-FA-2Opt
b. St70 best tour using AS-FA-2Opt
c. Eil76 best tour using AS-FA-2Opt
d. Rat99 best tour using AS-FA-2Opt
d. KroA100 best tour using AS-FA-2Opt
e. KroA200 best tour using AS-FA-2Opt

need for self-adaptation mechanisms for ACO solvers and confirms the pertinence of the proposed hybridization and other complex algorithmic schemes. The set of recommended parameters for ACO found by the FA are listed in Table 3 for each problem of the TSP instances used in this study.

AS-FA-Ls converges convergences in 50 iterations with a possible acceptable solution -an error of less then 1%. The best obtained solutions for large TSP instances like Kroa 200 needed 200 iterations. The Results presented in Table 3 are obtained with the maximum number of iterations fixed to 100. The obtained errors ranged from 1.58 to 0.

Figure 8 shows how the tour length which is also set to be the cost or fitness function of FA evolved while solving the berlin52 TSP instance.

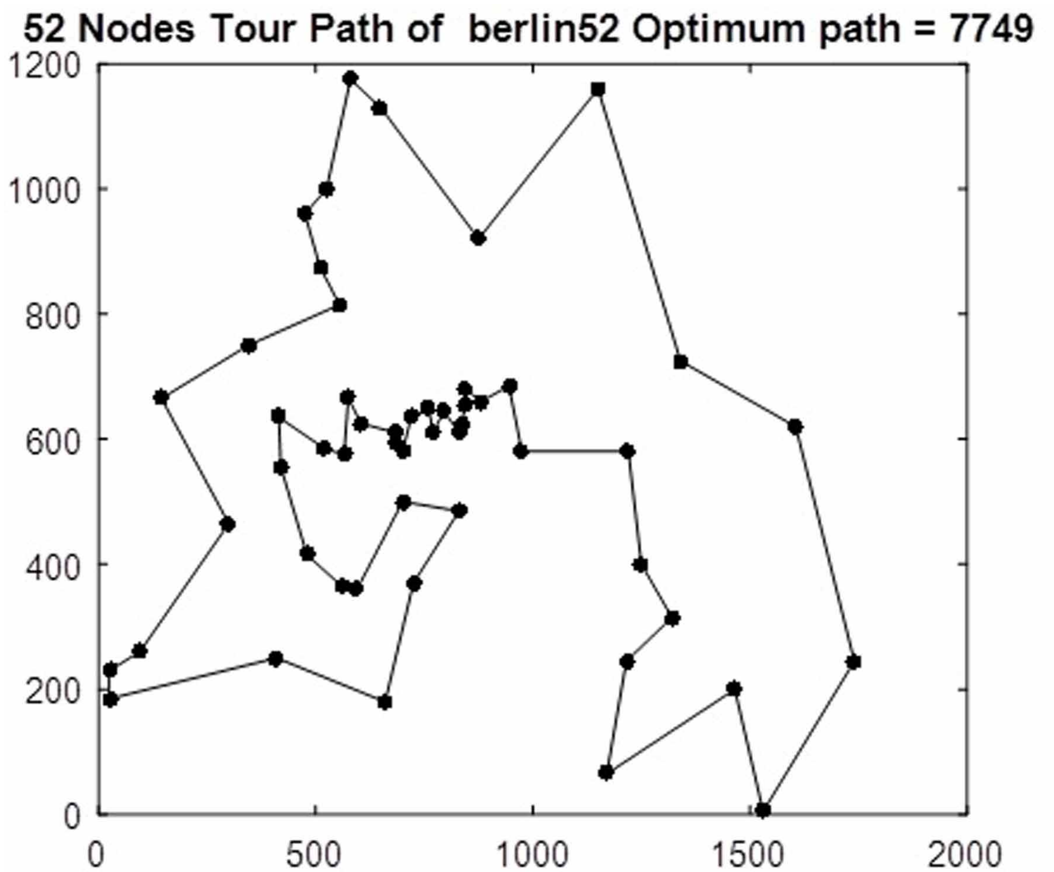
The investigations regarding the execution time of the proposed for the Berlin 52 TSP instance, showed that the method is relatively fast, and that the processing time for 10 FA search agents is about 277s, while it rises to 314s and 384s for 20 and 30 agents, respectively. The full results are shown in Table 4.

The processing time increases as the number of agents' increase, in a non-linear manner, since the increase in the number of agents also increases the possibility to find a good set of parameters allowing an early convergence.

CONCLUSION AND PERSPECTIVES

This study investigated a hybrid self-adaptive algorithm for an NP-Hard Combinatorial Optimization problem, the TSP. The proposed method, AS-FA-Ls have performance comparable to the state-of-art algorithms. Globally, AS-FA-2opt is achieving fair solutions for the TSP problems used in the experimental evaluation including berlin52, St70, eil76, KroA100, kroA200, etc.

Figure 6. berlin52 optimum tour obtained AS-FA-Ls

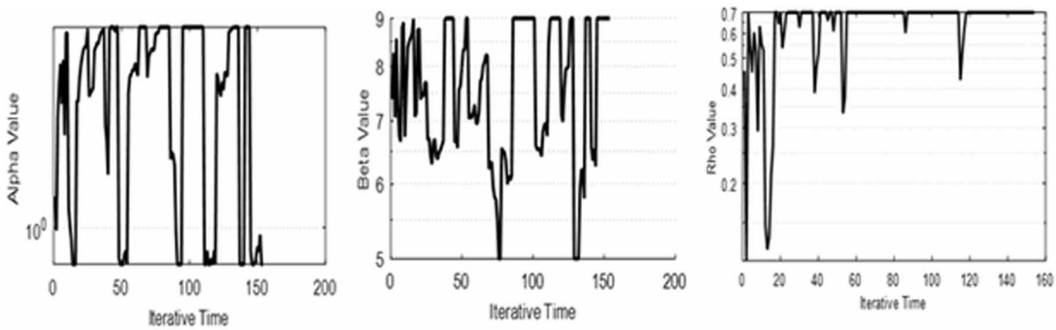


Clearly, AS-FA-Ls returned better results than the reference methods in the following cases: berlin52, st70, eil76, rat99, kroA100 and kroA200.

The analysis of the results allowed determining a set of recommended parameters to be used with ACO for the TSP instances used in this study.

Future investigations will focus on the impact of the local search strategy including similar techniques such as K-means, Greedy search, or Variable Neighborhood Search (VNS). The comparative evaluation of similar hybridizations based on bio-inspired and heuristic technique may also subject to investigations…

Figure 7. Convergence of ACO parameters (berlin52) using AS-FA-Ls



a. Convergence of Alpha (α) Value b. Convergence of Beta (β) Value c. Convergence of Rho (ρ) Value

Table 3. Optimized ACO parameters for each test bench (AS-FA-Ls)

Problems	±	²	*A*
eil51	7	9	0.03
berlin52	0.7	5	0.7
st70	7	0	0.096375
eil76	7	5	0.7
rat99	0.7	9	0.7
eil101	7	5	0.03
kroA100	0.7	9	0.7
ch150	0.7	9	0.7
kroA200	0.7	9	0.7

Figure 8. Evolution of the fitness function berlin52 using AS-FA-Ls

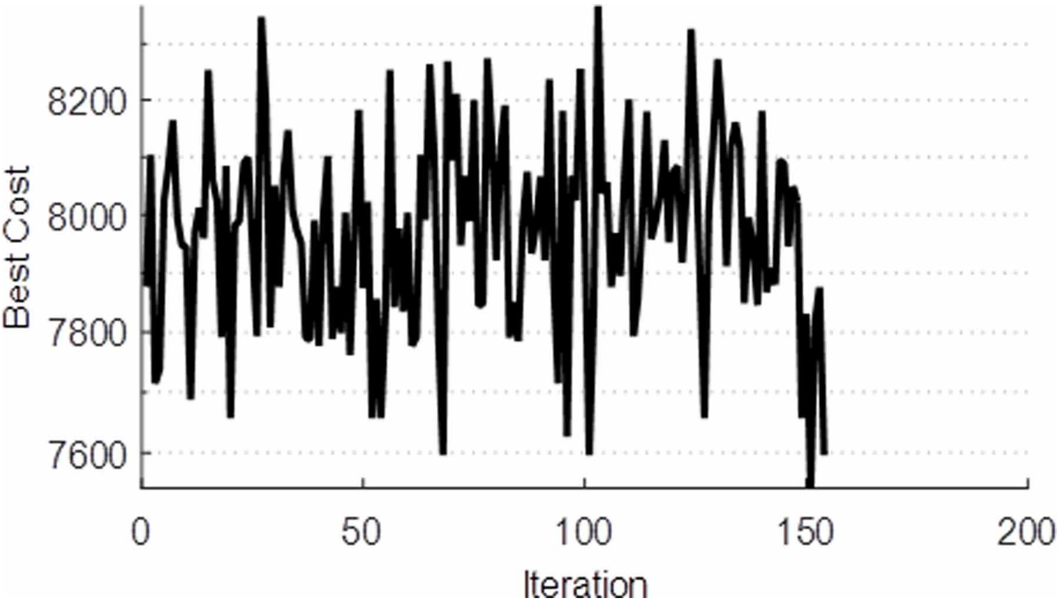


Table 4. Execution time for AS-FA-Ls

Maximum Iterations = 100	
Swarm size =10	**Execution time in (s)**
10	277.5886
20	314.5676
30	384.4568

REFERENCES

Ariyaratne, M. K. A., Fernando, T. G. I., & Weerakoon, S. (2018). A self-tuning firefly algorithm to tune the parameters of ant colony system. *International Journal of Swarm Intelligence*, *3*(4), 309–331. doi:10.1504/IJSI.2018.091415

Bajracharya, A. (2016). Public transportation and private car: A system dynamics approach in understanding the mode choice. *International Journal of System Dynamics Applications*, *5*(2), 1–18. doi:10.4018/IJSDA.2016040101

Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations Research*, *6*(6), 791–812. doi:10.1287/opre.6.6.791

Dorigo, M., & Birattari, M. (2007). Swarm intelligence. *Scholarpedia*, *2*(9), 1462. doi:10.4249/scholarpedia.1462

Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *Biosystems, 43*(2), 73-81.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization. 2004*. Massachusetts Institute of Technology. doi:10.7551/mitpress/1290.001.0001

Goel, R., & Maini, R. (2018). A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems. *Journal of Computational Science*, *25*, 28–37. doi:10.1016/j.jocs.2017.12.012

Gündüz, M., Kiran, M. S., & Özceylan, E. (2015). A hierarchic approach based on swarm intelligence to solve the traveling salesman problem. *Turkish Journal of Electrical Engineering and Computer Sciences*, *23*(1), 103–117. doi:10.3906/elk-1210-147

Helsgaun, K. (2006). *An effective implementation of K-opt moves for the Lin-Kernighan TSP heuristic* [Doctoral dissertation]. Roskilde University.

Hunt, B. R., Lipsman, R. L., & Rosenberg, J. M. (2014). *A guide to MATLAB: for beginners and experienced users*. Cambridge university press. doi:10.1017/CBO9781107338388

İlhan, İ. (2017). 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), 332–345. doi:10.1080/08839514.2017.1339983

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Erciyes University.

Kefi, S., Rokbani, N., Krömer, P., & Alimi, A. M. (2016, October). Ant supervised by PSO and 2-opt algorithm, AS-PSO-2Opt, applied to traveling salesman problem. In *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 4866-4871). IEEE Press. doi:10.1109/SMC.2016.7844999

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization (PSO). In *Proceedings of the IEEE International Conference on Neural Networks* (pp. 1942-1948). Academic Press. doi:10.1109/ICNN.1995.488968

Kumbharana, S. N. & Pandey, G. M. (2013). Solving travelling salesman problem using firefly algorithm. *International Journal for Research in science & advanced Technologies, 2*(2), 53-57.

Laporte, G., & Martello, S. (1990). The selective travelling salesman problem. *Discrete Applied Mathematics*, *26*(2-3), 193–207. doi:10.1016/0166-218X(90)90100-Q

Mahi, M., Baykan, Ö. K., & Kodaz, H. (2015). A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem. *Applied Soft Computing*, *30*, 484–490. doi:10.1016/j.asoc.2015.01.068

Martinez, W. L., & Martinez, A. R. (2015). *Computational statistics handbook with MATLAB*. Chapman and Hall/CRC.

Mousa, M. E., Ebrahim, M. A., & Hassan, M. M. (2015). Stabilizing and swinging-up the inverted pendulum using PI and PID controllers based on reduced linear quadratic regulator tuned by PSO. *International Journal of System Dynamics Applications*, *4*(4), 52–69. doi:10.4018/IJSDA.2015100104

Pradhan, P. L. (2017). Proposed Heuristics Model Optimizing the Risk on RTS. *International Journal of System Dynamics Applications*, *6*(2), 31–51. doi:10.4018/IJSDA.2017040102

Rajasekaran, V., Aranda, J., & Casals, A. (2014). Recovering planned trajectories in robotic rehabilitation therapies under the effect of disturbances. *International Journal of System Dynamics Applications*, *3*(2), 34–49. doi:10.4018/ijsda.2014040103

Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA Journal on Computing*, *3*(4), 376–384. doi:10.1287/ijoc.3.4.376

Rokbani, N., Abraham, A., & Alimi, A. M. (2013). Fuzzy ant supervised by PSO and simplified ant supervised PSO applied to TSP. In *Proceedings of the 13th International Conference on Hybrid Intelligent Systems (HIS 2013)* (pp. 251-255). IEEE. doi:10.1109/HIS.2013.6920491

Rokbani, N., Abraham, A., Twir, I., & Haqiq, A. (2019). 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. *International Journal of Hybrid Intelligent Systems*, *15*(1), 17–26. doi:10.3233/HIS-180258

Rokbani, N., Casals, A., & Alimi, A. M. (2015). IK-FA, a new heuristic inverse kinematics solver using firefly algorithm. In *Computational Intelligence Applications in Modeling and Control* (pp. 369–395). Cham: Springer. doi:10.1007/978-3-319-11017-2_15

Rokbani, N., Kromer, P., Twir, I., & Alimi, A. M. (2019). A new hybrid gravitational particle swarm optimisation-ACO with local search mechanism, PSOGSA-ACO-Ls for TSP. *International Journal of Intelligent Engineering Informatics*, *7*(4), 384–398. doi:10.1504/IJIEI.2019.101565

Twir, I., Rokbani, N., Haqiq, A., & Abraham, A. (2017, December). Experimental Investigation of Ant Supervised by Simplified PSO with Local Search Mechanism (SAS-PSO-2Opt). In *Proceedings of the International Conference on Soft Computing and Pattern Recognition* (pp. 171-182). Springer.

Wahid, F., Ghazali, R., & Shah, H. (2018, February). An improved hybrid firefly algorithm for solving optimization problems. In *Proceedings of the International conference on soft computing and data mining* (pp. 14-23). Springer. doi:10.1007/978-3-319-72550-5_2

Yang, X. S. (2009), Firefly algorithms for multimodal optimization. In *Proceedings of the International symposium on stochastic algorithms* (pp. 169-178). Springer.

Yang, X. S. (2010). Levy flights and global optimization. In *Research and development in intelligent systems XXVI* (pp. 209–218). London: Springer. doi:10.1007/978-1-84882-983-1_15

Nizar Rokbani earned an Electrical Engineering degree from the National Engineering School of Tunis, ENIT, 1995; a Master and a PhD in Electrical Engineering from the National Engineering School of Sfax, ENIS respectively in 2003 and 2013. He worked as automation & process engineer until 1998, then as a technology trainer at the Institute of Technological Studies of Gabes, occupied a post of training adviser engineer at the National Agency of Vocational Training from 2003 to 2006. In 2006 he joined the University of Gabes as an assistant lecturer, and then moved to the University of Sousse. Dr. Rokbani is an Assit Prof of industrial computing at the institute of applied science and technologies of Sousse since 2014. Dr. Nizar Rokbani is an IEEE Senior member; he served in several volunteering positions in IEEE RAS, SMC and OES Tunisia Chapters, and used to be the treasurer of Tunisia section (2015-2016). His research interests include applications of intelligent techniques such as Swarm intelligence, computational intelligence, fuzzy logic, evolutionary algorithms to robotic systems and industrial processes.

Pavel Kromer graduated in Computer Science at FEECS. He worked as an analyst, developer, and trainer in a private company between 2005 and 2010. Since 2010, he has worked at the Department of Computer Science, FEECS. In 2014, he was a Postdoctoral Fellow at the University of Alberta. In 2015, he was awarded the title Assoc. Professor of Computer Science. He was Researcher at the IT4Innovations (National Supercomputing Center) between 2011 and 2016 and has been a member of its scientific council in 2017. Since September 1, 2017, he has been the Vice Dean for International Cooperation at FEECS. Since 2018, he is a Senior Member of the IEEE. In his research, he focuses on computational intelligence, information retrieval, data mining, machine learning, soft computing and real-world applications of intelligent methods.

Ikram Twir Ikram Twir is a software engineer working in an international Company in Tunisia, eXo Platform. Ikram Touir is a graduate student from Tunisia in the field of artificial intelligence. She has been an active IEEE member and volunteer since 2014 when she established the IEEE Student Branch at Higher Institute of Applied Science and Technology of Sousse -Tunisia. In 2018,, she received her research master degree in Intelligent Pervasive Systems. Her current research interests include Artificial Intelligence, Computational Intelligence, Evolutionary Algorithms, Bio-inspired Algorithms, Internet of Things, and Robotics.

Adel M. Alimi (IEEE Student Member'91, Member'96, Senior Member'00). He graduated in Electrical Engineering in 1990. He obtained a PhD and then an HDR both in Electrical & Computer Engineering in 1995 and 2000 respectively. He is full Professor in Electrical Engineering at the University of Sfax, ENIS (National Engineering School of Sfax), since 2006. He is founder and director of the research REGIM-Lab. in intelligent Machines. He is director of the Tunisia Erasmus+ Office, since 2018. He is an active IEEE volunteer, served and founded several IEEE technical chapters in Tunisia, including SMCs, RAS, CIS, OES….. He is the Chair of IEEE Tunisia section (2019-2020). Prof Alimi is the author of more than 100 Journal papers and more than 600 conference papers, his research interests covers artificial intelligence theory and applications, with emphasis on pattern recognitions, neural networks, Swarm intelligence, deep learning and robotics.