Applied Soft Computing 25 (2014) 234–241

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP

Walid Elloumi [a,*], Haikal El Abed [a,b], Ajith Abraham [a,c], Adel M. Alimi [a]

[a] REGIM-lab: Research Groups on Intelligent Machines, University of Sfax, National Engineering School of Sfax (ENIS), BP 1173, Sfax 3038, Tunisia
[b] Technical Trainers College (TTC), German International Cooperation (GIZ), P.O. Box 2730, Riyadh 11461, Saudi Arabia
[c] Machine Intelligence Research Labs (MIR Labs), P.O. Box 2259, Auburn, WA 98071-2259, USA

ARTICLE INFO

Article history:
Received 4 July 2012
Received in revised form
11 September 2014
Accepted 18 September 2014
Available online 28 September 2014

Keywords:
Swarm intelligence
Ant colony optimization
Particle swarm optimization
Traveling salesman problem
Multi-objective optimization

ABSTRACT

Swarm-inspired optimization has become very popular in recent years. Particle swarm optimization (PSO) and Ant colony optimization (ACO) algorithms have attracted the interest of researchers due to their simplicity, effectiveness and efficiency in solving complex optimization problems. Both ACO and PSO were successfully applied for solving the traveling salesman problem (TSP). Performance of the conventional PSO algorithm for small problems with moderate dimensions and search space is very satisfactory. As the search, space gets more complex, conventional approaches tend to offer poor solutions. This paper presents a novel approach by introducing a PSO, which is modified by the ACO algorithm to improve the performance. The new hybrid method (PSO–ACO) is validated using the TSP benchmarks and the empirical results considering the completion time and the best length, illustrate that the proposed method is efficient.

1. Introduction

Swarm intelligence, was born from the incredible abilities of social insects to solve their problems [1]. Their colonies incorporating either few animals or millions of individuals show fascinating behaviors combining the efficiency of the flexibility and robustness [2].

Swarm based optimization algorithms could be applied in different fields like distributed control of collective robotics or network traffic management [3–5], etc. Building of efficient structures for dynamic task allocation is also a possible application of ACO [6]. In [7] some examples of complex and sophisticated behaviors are listed and the diversity of social insects optimization paradigms are discussed. The roots of swarm intelligence are set so deeply in the study of self-organized behavior in social insects. From the routing of traffic in telecommunication networks to the design of control algorithms for groups of autonomous robots, collective behavior of these animals are the source of inspiration for many works belonging to this emerging area of research [33].

Swarm intelligent methods are part of the meta-heuristic [9,20] based algorithm family. A heuristic method is an iterative technique, that reproduces a natural process of physical, chemical or biological system with self organization and evolving capacities. Like in nature, it can search local optimal solutions or global optimal solution using simple rules. In fact the result will depend partially on the problem that we are solving and execution time limit (number of iterations or stopping condition). Hence metaheuristics are usually reserved for difficult optimization problems. Many heuristic search methods have been used in a cooperative search environment including Tabu Search (TS) [14], Genetic algorithms (GA) [15,16], Ant colony optimization [17,26,28] and Particle swarm optimization [18,36]. For all these methods, the challenging issue is the choice of their parameters, for example in PSO, the number of iterations, number of particles and the choice of the fitness function are the key elements that control the exploration of the search space.

Ouyang and Zhou have inserted the delete-crossover strategy into an improved PSO–ACO to tackle the problem of PSO–ACO to solve large scale TSP. The improvement of the performance of PSO–ACO versus the ACO was presented in [34]. Two popular swarm inspired methods in computational intelligence are ant colony optimization and particle swarm optimization. The PSO is simple and promising, and it requires less computation time, though it faces difficulties for solving discrete optimization problems [10,11]. Inspired by the food-seeking behavior of real ants,

* Corresponding author. Tel.: +216 55 556 336; fax: +216 74 677 545.
E-mail addresses: walid.elloumi@ieee.org, elloumiwalid@gmail.com (W. Elloumi), elabed@ieee.org (H. El Abed), ajith.abraham@ieee.org (A. Abraham), adel.alimi@ieee.org (A.M. Alimi).

http://dx.doi.org/10.1016/j.asoc.2014.09.031
1568-4946/

ant systems [13], have demonstrated to be an efficient and effective tool for combinatorial optimization problems. We deal with the main tenets that underlie in the organization of insects' colonies and swarms. We start with recalling some notions of the decentralized nature of these systems and detect the underlying mechanisms of complex collective behaviors of social insects, the notion of stigmergy and the theory of self-organizing systems [33]. We note the role of interactions and the importance of bifurcations that can be observed in the hybrid optimization by ACO and PSO [12,37,42]. This paper is organized as follows. Section 2 gives an overview of accomplishments of the social insects in general. Different problems solving devices inspired by the collective behavior of insect colonies and flock of birds that are introduced using computational swarm intelligence are proposed in Section 3. Section 4 presents hybrid PSO and ACO when we have presented our proposal PSO modified by ACO. Section 5 presents the comparative study when using TSP problem. The paper is concluded in Section 6.

2. Accomplishments of the social intelligence

The optimization potential of simple behaviors has been most noted in studies of insects, and in particular in the behaviors of the social insects. An insect may have only a few hundred brain cells, but insect organizations are capable of architectural marvels, elaborate communication systems, and terrific resistance to the threats of nature. Insects develop complex social relationships and interactions based on the elementary intelligence capacities of each member of the group, these social compartments create a set of higher intelligent skills, called social intelligence. Similar attitudes are also observed in bird flocks or fish bands.

The construction of mass behaviors from the behaviors of single ants is the central problem in the sociobiology of insects. Given that the behavior of a single ant is almost random, with a stochastic tendency to gravitate toward paths that have been trodden by other ants; the achievements of swarms of ants are most incredible. An isolated ant's behavior quickly results in the demise of the individual, but the mass behavior of a colony of ants provides sustenance and defensive protection for the entire population. The extrapolating descriptions of the fixed action patterns and the stereotypical instinctive behaviors of organisms, theorize that the dazzling accomplishments of ant societies could be explained and understood in terms of simple fixed action patterns, which includes behavioral responses to pheromone [21].

The study of complex systems and the rise to prominence of computer simulation models of such systems have given scientists the tools they need to model the simple behaviors of ants and how they can combine to produce an effect that is much more than the sum of its parts, and these insights have in turn led to more insights about the nature of man and society and about the physical world [8]. Insect sociability is a classic example of the emergence of global effects from local interactions [25].

In any social organization there is some degree of communication among the members, just enough to keep them close to each others and to ensure to global harmonious comportment, any similar social group can be considered as a swarm. By this minimal communication they can remind each other that they are not alone but are cooperating with team mates. Assuming that an ant colony can be thought as a swarm whose individuals are ants, a flock of birds is a swarm whose agents are birds, traffic is a swarm of cars, a crowd is a swarm of people, an immune system is a swarm of cells and molecules, and an economy is a swarm of economic agents.

ACO is used in industrial environments where computational resources and time are limited. This paper sheds the light on modifying ACO using two operations: first by adjusting the parameter Q_0, which relates to both exploitation and exploration in the local search of ACO. Second, escaping the trap by reinitializing Q_0 in a way of exploration. The results have shown that the modifications brought to ACO surpass over the traditional ACO in terms of accuracy rate and iteration utilization [35].

A combination of two well-known meta-heuristic algorithms PSO and Ant colony optimization (ACO) is realized. The latter is based on a frame work design named A–B - domain which is a design that combines two heuristic algorithms A and B hierarchically and Domain is the set of feasible sets of parameters for algorithm B working on the problem [19].

3. Basics of ACO and PSO

3.1. Ant colony optimization

Colorni [22] showed how a very simple pheromone following behavior could be used to optimize the traveling salesman problem [23]. Their ant colony optimization was based on the observation that ants would find the shortest path around an obstacle separating their nest from a target such as a piece of candy simmering on a summer sidewalk.

As ants move around they leave pheromone trails, which dissipate over time and distance. The pheromone intensity at a spot, that is, the number of pheromone molecules which a wandering ant might encounter, is higher either when ants have passed over the spot more recently or when a greater number of ants have passed over the spot. Thus, ants following pheromone trails will tend to congregate simply from the fact that the pheromone density increases with each additional ant that follows the trail.

Dorigo [21] focused on the fact that ants meandering from the nest to the candy and back will return more quickly, and thus will pass the same points more frequently, when following a shorter path. Passing more frequently, they will lay down a denser pheromone trail. As more ants pick up the strengthened trail, it becomes increasingly stronger In their computer adaptation of these behaviors, Dorigo et al. let a population of ants search a traveling salesman map stochastically, increasing the probability of following a connection between two cities as a function of the number of other simulated ants that had already followed that link. By exploitation of the positive feedback effect, that is, the strengthening of the trail with every additional ant, this algorithm is able to solve quite complicated combinatorial problems where the goal is to find a way to accomplish a task in the fewest number of operations.

Research on live ants has shown that when food is placed at some distance from the nest, with two paths of unequal length leading to it, they will end up with the swarm following the shorter path. If a shorter path is introduced, though, for instance, an obstacle is removed, they are unable to switch to it. If both paths are of equal length, the ants will choose one or the other. If two food sources are offered, with one being a richer source than the other, a swarm of ants will choose the richer source; if a richer source is offered after the choice has been made, most species are unable to switch, but some species are able to change their pattern to the better source. If two equal sources are offered, an ant will choose one or the other arbitrarily.

They compare the communications network within a swarm of ants which can be described in terms of three characteristics:

- their structure comprises a set of nodes and their interconnections,
- the states of node variables change dynamically over time,
- there is learning changes in the strengths of the connections among the nodes.

Ants follow a random movement as there is no systematic pattern pheromone; the latter depends on two parameters: the strength of the pheromone and its attractiveness to ants. If the distribution of pheromone is variable, or if the attraction of ants to the pheromone is low, then no model can be formed. On the other hand, if a high pheromone concentration is constituted, or if the attraction of ants to the pheromone is very intense, we can see an emerging sub-optimal model, as the ants come together in a kind of unnecessary compliance. At the very edge of chaos where the parameters are tuned correctly, the ants will explore and follow the pheromone signals, and wander from the swarm, and come back to it, and eventually coalesce into a pattern that is, most of the time, the shortest, most efficient path from here to there [32].

From simulated and real examples of achievements of insects, we can see the optimization of various types, depending on clustering elements or finding the shortest path through a landscape, with some interesting features. None of these cases include the overall assessment of the situation: an insect can detect its immediate environment. The optimization requires a method to assess the ability of a solution, which appears to require that candidate solutions must be compared to a standard which may be a desired end state or the fitness of alternatives. The bottom-up methods of insect societies, however, do not permit evaluation in fact-no ant knows that swarm. Generally, the communication method means that pheromone pathway will be more attractive, with an autocatalytic accumulation of pheromone resulting from the convergence of the population on the behavior of all the most-fit locally [32].

Based on our construction graph definition, the algorithm encompassing different well-known versions of Ant System [1,21–23] (see Fig. 1).

They begin by initializing the ants on a start node. For each ant it will choose the next node, if there are other nodes to visit we return to step 2. If not they go back to the first node and the amount of pheromone is changed during a cycle. If the stopping criterion is successful best lap is displayed, otherwise it returns to step 2. The main elements of the above process are: artificial ants, simple computational agents that individually and iteratively construct solutions for the problem, which has been modeled as a graph. Ants explore the graph visiting nodes connected by edges. A problem solution is an ordered sequence of nodes. The search process is executed in parallel over several constructive computational threads. A dynamic memory structure, inspired by the pheromone laying process which incorporates information on the effectiveness of previously obtained results, guides the construction process of each thread. The intermediate partial problem solutions are seen as state at each iteration k of the algorithm each ant move.

3.2. Particle swarm optimization

Kennedy and Eberhart [10], have studied the comportment of social animals trying to understand how these groups attend their global goals while performing individual comportment. Early in 1995 they made public a first attempt to describe how this kind of social intelligence rose from simple and implicit rules. A simplified numerical model is then introduced using (5). In this formal model the swarm is composed by a set of particles (Eq. (1)), if p represents a member, its proprieties are limited to its coordinates and its velocity as given in (2) and (4).

A particle moves according to its own position with respect to its best neighborhood and to the best global location of the swarm [24]. The best number is selected on the basis of a limited search space and a fitness function. The best global location is selected from the best local locations as the one with a better fitness value, maximizing or minimizing the cost depending on the problem that the algorithm is solving. The system iterates but with a fixed number of iterations known before launching the search process, that ensures to PSO to converge to a solution, even if this one is not the best one.

The traditional PSO iterates as follows. First of all, we have to decide how many particles we need to use to solve the problem. Every particle has its own position, velocity and best solution. For example, let S be an n dimensional search space, $f : S \to \mathbb{R}$ and N the number of particles that comprise the swarm,

$$S = \{x_1, x_2, \ldots, x_n\} \tag{1}$$

Then the i th particle is a point in the search space,

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^t \in S, \tag{2}$$

as well as best position,

$$p_i = (p_{i1}, p_{i2}, \ldots, p_{in})^t \in S, \tag{3}$$

which is the best position ever visited by x_i, during the search. The velocity of x_i; is also an n dimensional vector,

$$v_i = (v_{i1}, v_{i2}, \ldots, v_{in})^t \tag{4}$$

The process of velocity update is given below, where c_1 and c_2 are constants, r_1 and r_2 are random variables in the range from 0 to 1. $p_{il}(t)$ is the best local solution of the i th particle for the iteration number up to the i th iteration and the $p_{ig}(t)$ is the best global solution of all particles. A particle velocity should be updated using the following equation:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 (p_{il}(t) - x_{ij}(t)) + c_2 r_2 (p_{ig}(t) - x_{ij}(t)) \tag{5}$$

To prevent the velocity from being too large, we set a maximum value to limit the range of velocity,

$$-v_{\max} \le v \le v_{\max} \tag{6}$$

Further, the particle is moved according to:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)\; 1 \le i, j \le N \tag{7}$$

The best local solution and the best global solution are selected according to a fitness function that should fit the proposed problem. These iterative steps will go on unless we reach the termination condition. Find next diagram representing the original PSO proposal. Fig. 2 The process of particle swarm optimization, Lp: Loop condition (*ni* iteration = *nmax*) or (fitness satisfied).

4. Hybrid ACO and PSO

4.1. Hybrid optimization of PSO and ACO

Elloumi et al. introduced the combination between Ant colony optimization and particle swarm optimization [12]. Applied to traveling salesman problem, it consists of building many functions seeking to optimize the path of the TSP depending on the selected policy. Policy can be for example: minimization of the global distance, minimization of the travel cost, etc. In the following paragraphs the proposal will be applied to the TSP with respect of the first policy.

PSO, on the other hand, constitutes a crucial element to serve the purpose of our study. It includes mainly the algorithm variations. On the next stage PSO will be coupled with ACO to achieve the combinatorial optimization.

We have proposed an ant colony algorithm supervised by Particle Swarm optimization to solve continuous optimization problems [27]. Traditional ACO is used for discrete optimization while PSO is for continuous optimization problems and together PSO and ACO have shown a great potential in solving a wide range of optimization problems [12,42].

PSO consists of three stages, namely, the positions and velocities of particle production, updating speed, and finally, updating

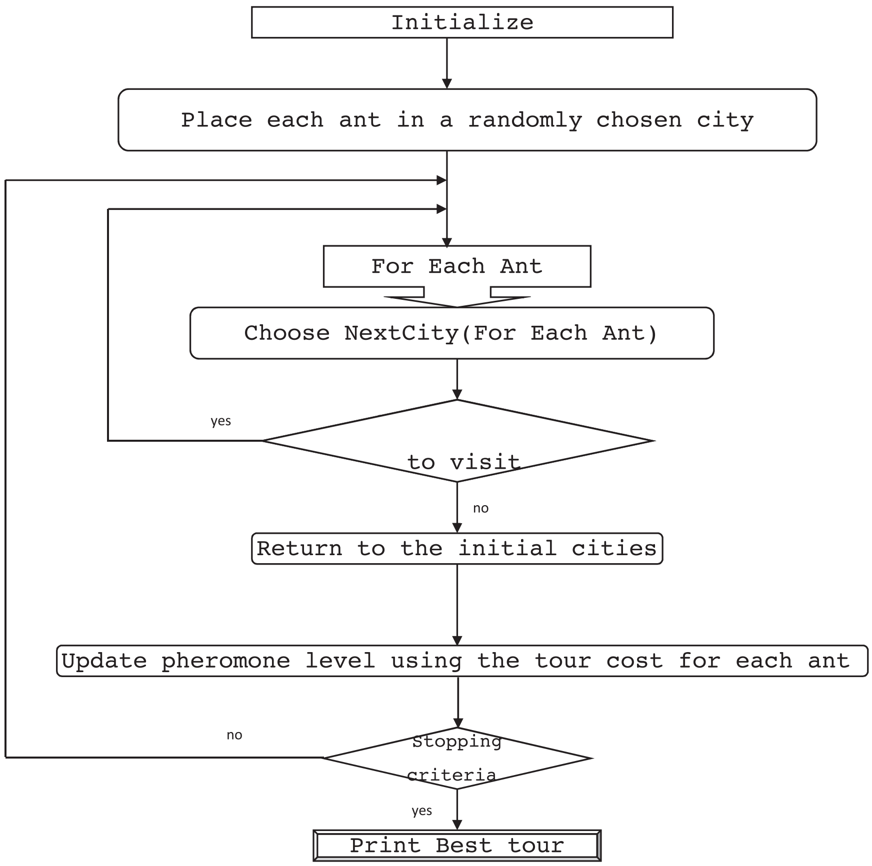

Fig. 1. Ant system algorithm for TSP.

position. PSO is easy to implement in computer simulations using mathematical operations and basic logic, since its working mechanism involves only two basic rules update. The particles are conceptual entities that constitute a swarm flying through space multidimensional search. The relationship between the swarm of particles in PSO is similar to the relationship between population and chromosomes in the genetic algorithm. At any given time, each particle has a position and a velocity. The position vector of a particle relative to the origin of the search space is a test solution to the problem of search. These particles fly with a certain velocity and find the best global position after some iteration. At each iteration, each particle can adjust its velocity vector, based on its momentum and the influence of his best position (pbest – particle best) and the best position of its neighbors (gbest – global best), then calculate a new position that "particle" is to fly. In other words, it finds the global optimum by simply adjusting the trajectory of each individual toward its own best location and the best particle swarm with each generation of evolution. The direction of the particle swarm is defined by the set of neighboring particles of the particle and its historical experience.

4.2. PSO modified by ACO

Details about ACO and PSO can be found respectively in Sections 3.1 and 3.2. Where τ_{ij}, represents the quantity of pheromone of ant_{ij}, depends on the following probability:

$$P_{ij}^k(t) = \begin{cases} \dfrac{(\tau_{ij}(t)^{\alpha})(\eta_{ij})^{\beta})}{\sum_{i \in J_i^k} (\tau_{ij}(t)^{\alpha})(\eta_{ij})^{\beta}} & \text{If}(i,j) \in J_i^k \\ 0 & \text{If}(i,j) \notin J_i^k \end{cases} \tag{8}$$

where the values η_{ij} are called heuristic information values, that we get through some problem-specific heuristic. The quantities $\eta_{ij}(u)$ may depend on the entire partial path u traversed. The function g blends pheromone trail heuristic information; hence,

$$g(\tau, \eta) = \tau^{\alpha} \cdot \eta^{\alpha} \tag{9}$$

where $\alpha > 0$ and $\beta > 0$.

The pheromone trails are real values transferred to the edges of the construction graph. They are deemed as the memory of the procedure strong information on how good the single solution components have turned out in previous iterations. Knowing that τ_{ij} is a time depending quantity, the particle corresponding to an attractive line joining two nodes and confirmed by ants will be accelerated, since their velocities will increase with the term τ_{ij}. At the opposite the particle corresponding to ants with week search results, will receive the classical velocity of the swarm, since the term τ_{ij} will be equal to 0.

We deduce that after finishing its cycle each ant leaves certain quantities of pheromone $\Delta_{ij}^k(t)$ which depends on

$$\Delta_{ij}^k(t) = \begin{cases} \dfrac{Q}{L^k(t)} & \text{If}(i,j) \in T^k(t) \\ 0 & \text{If}(i,j) \notin T^k(t) \end{cases} \tag{10}$$

where $T^k(t)$ is the target taken by the ant k for iteration t, $L^k(t)$ is the length of tour and Q is the fixed parameter.

The Pheromone deposit by the ants' mechanisms would be used by the PSO as a weight of its particles ensuring a better global search strategy referring to Eq. (5)+(Q/($L^k(t)$)). By using the PSO modified by ACO (PSO-M-ACO) design method, the hybrid PSO–ACO in the feasible domain can explore their chosen regions rapidly and

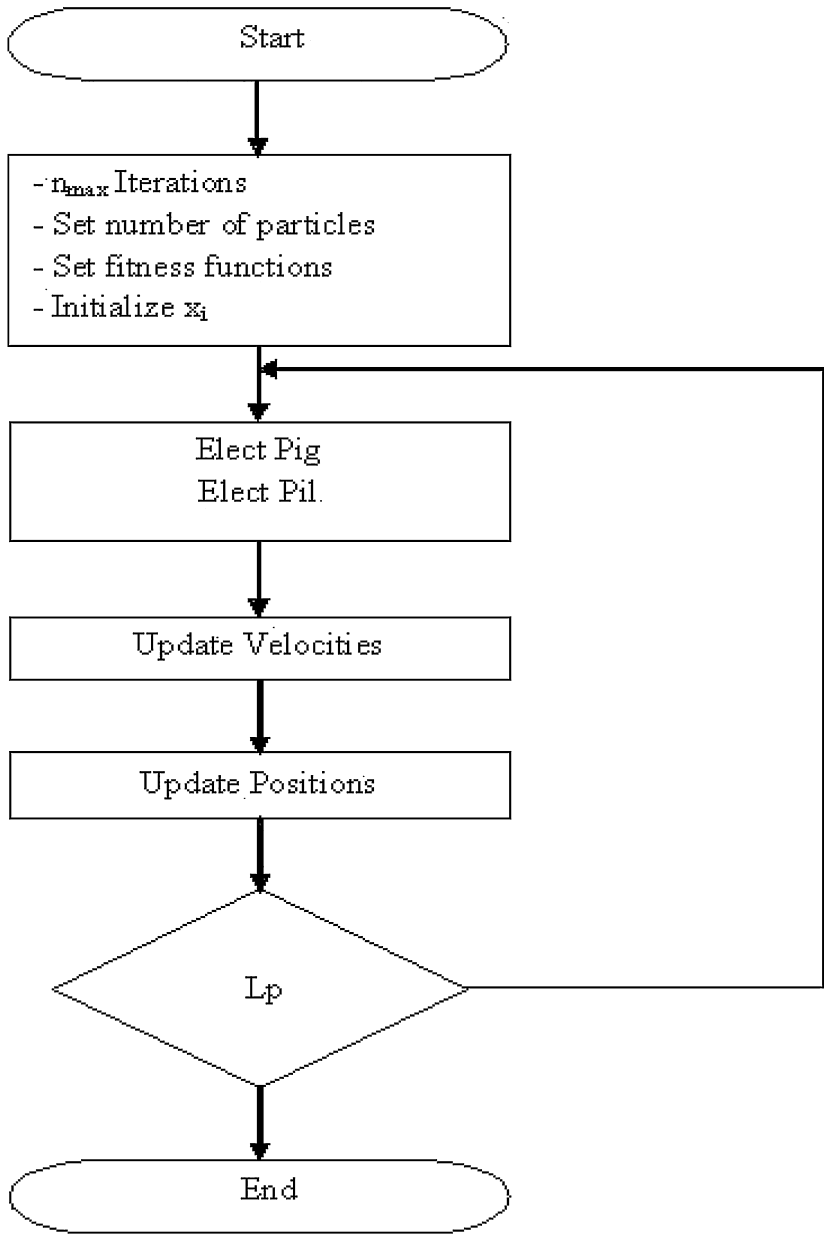

Fig. 2. The process of particle swarm optimization, Lp: Loop condition (n_i iteration = n_{max}) or (fitness satisfied).

efficiently because the best ant will be the global particle in PSO while the local remains the best regarding its neighbors.

A closer examination of this concept will be given throughout our study. To reach our aim, we have to carry out different steps. First, a general framework of the system of nodes that stands for different functions and the set of the nodes which translate these functions and constitute the entire charts representing localizations is required. Second, a study of the execution time and the shortest path established is needed. All that is shown in Fig. 3.

5. Comparative study

We describe the TSP and compare the results obtained with the same configuration of the problem with three proposals.

5.1. TSP presentation

The traveling salesman problem is a classical example of a combinatorial optimization problem, which has proved to be NP-hard. In the TSP, the objective is to find the salesman's tour to visit all the N cities on his list once and only once, returning to the starting point after traveling the shortest possible distance. Additionally, we assume that the distance from city i to city j is the same as from city j to city i (symmetrical TSP). A tour can be represented as an ordered list of N cities. In this case, for $N > 2$ there is $N!/2N$ different tours (the same tour may be started from any city from among N cities and traversed either clockwise or anti-clockwise). Many methods are used for solving the TSP, e.g.: the Lin-Kernighan algorithm, simulated annealing [29]. TSP can be solved using neural networks. A self-organizing neural network [30] and the Hopfield network are able to solve the TSP [31].

However, most of these algorithms do not tackle large instances. Previously reported results on different approaches to solve TSP, such Artificial Bee Colony Algorithm [38,39] or Genetic Algorithm for TSP [40,41], show the robustness and efficiency of the PSO and ACO algorithms during the variation of instances. This lead us to investigate in the comparison between hybrid PSO and ACO in our approach.

Table 1
Comparison between hybrid PSO and ACO.

Nodes	Population	T.PSO-M-ACO	L.PSO-M-ACO	T.ACO	L.ACO	T.PSO	L.PSO
22	22	**4.6486e−004**	**76.1195**	0.2438	77.8000	0.1406	90.6884
	70	**4.9084e−004**	**76.0570**	0.4969	77.1834	0.4556	90.4220
	100	**5.1180e−004**	**75.3097**	0.6866	77.0269	0.5707	89.4898
29	29	**5.2716e−004**	**9.3364e+003**	0.4190	1.1621e+004	0.4181	1.1761e+004
	70	**5.3247e−004**	**9.3294e+003**	1.0881	1.0677e+004	0.9632	1.0900e+004
	100	**5.6208e−004**	**9.2746e+003**	1.3713	1.0432e+004	1.2177	1.0472e+004
30	30	**5.5622e−004**	**480.0967**	0.3724	539.0477	0.3577	584.0341
	70	**5.4783e−004**	**459.3399**	0.7652	495.5985	0.7334	562.0160
	100	**5.6655e−004**	**454.6212**	1.4225	491.7651	1.1059	545.7844
48	48	**6.7243e−004**	**3.5651e+004**	4.0005	4.2086e+004	3.0961	4.5973e+004
	70	**7.7161e−004**	**3.4606e+004**	4.5047	4.1420e+004	4.3249	4.4654e+004
	100	**7.0903e−004**	**3.4271e+004**	6.8139	4.1001e+004	4.8480	4.1158e+004
52	52	**7.6686e−004**	**7.9129e+003**	5.5777	8.9425e+003	4.0960	1.0404e+004
	70	**7.5093e−004**	**7.8692e+003**	5.5446	8.5578e+003	5.5387	1.0191e+004
	100	**6.9394e−004**	**7.542e+003**	8.1787	8.0068e+003	7.9181	1.0099e+004
70	70	**9.6716e−004**	**685.7039**	14.2154	755.4880	15.9345	910.6275
	100	**9.2330e−004**	**680.5641**	28.6257	734.5480	27.2070	848.9480
	150	**8.8838e−004**	**675.1635**	35.3854	697.3874	24.8191	845.9768
76	76	**9.4565e−004**	**568.2175**	15.2873	654.4432	14.6599	667.3720
	100	**9.4453e−004**	**550.7723**	19.1720	650.7820	18.9729	655.9789
	150	**9.2945e−004**	**538.0973**	33.1702	551.9371	28.8655	653.8607
96	96	**0.0011**	**582.7149**	47.1105	666.5597	47.0468	703.4219
	100	**0.0012**	**575.4039**	48.4213	631.4399	40.4907	671.3486
	150	**0.0011**	**575.0893**	63.7178	626.8571	61.0774	673.1950

5.2. Experimental results

In Table 1, T.PSO-M-ACO: the best Time of PSO modified by ACO (per seconds), L.PSO-M-ACO: the best length PSO-M-ACO, T.ACO: the best Time for ACO (per seconds), L.ACO: the best Length for ACO, T.PSO: the best Time for PSO (per seconds), L.PSO: the best Length for PSO.

We have used 1000 iterations because when we have tested 2000 and 3000 iterations we have concluded that the cycle number and time are proportionate to that of 1000 iterations. Due to this reason our combination consists in using 1000 iterations for the sake of time.

Our approach PSO-M-ACO is coded in Matlab and run on an Intel(R)P(R)1.7 GHz PC with 512 MB memory. There are many parameters used for the PSO-M-ACO. The Size of population, which we will increase three times, is the number of nodes of the social and cognitive probabilities, having c_1 and c_2, set as $c_1 = c_2 = 2$. while the inertia weight w is taken as 0.9, and the maximum of velocity v is taken as 100 and dimension of space as 10. Both α and β control the relative significance of pheromone trail and distance between cities in TSP where $\alpha = 1.5$, $\beta = 2$. ρ refers to the rate of pheromone evaporation $\rho = 0.7$. Each TSP run is conducted for 5 replications and 1000 iterations.

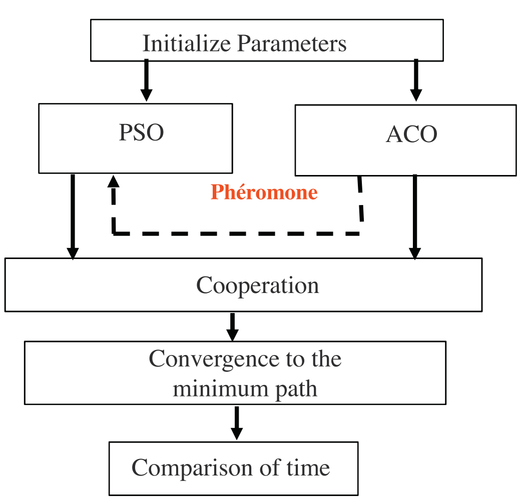

Fig. 3. PSO modified by ACO.

Table 1 summarizes the new improved results and comparison between PSO-M-ACO, ACO and PSO to the problem of TSP.

In fact, we notice that, as the size of population increases, execution time of the hybrid method decreases when compared to the time taken by PSO and ACO. Besides, as the size of population increases, the distance becomes shorter for the hybrid method. As evident from Table 1, we see that the execution time of the PSO is better compared to the ACO which is normal since the PSO is faster than ACO, but for the length of the shortest path to the PSO is poor compared to that of the ACO. This proves the need of the amount of pheromone deposited by ant, which was implemented in PSO, by the so called PSO-M-ACO. Our approach also finds the best solution and the best execution time as shown in Table 1.

Table 2 illustrates some comparisons with four heuristic methods and some of the benchmarks we have used. The algorithms used are Genetic algorithm (GA), Ant colony system (ACS), Ant System (AS) and the fuzzy ant colony optimization (FACO). If we compare our approach with respect to the proposed findings; we find that our approach converges rapidly to a minimum as the number of particles is increased in the swarm. This proves the necessity of hybridization used between PSO and ACO.

Fig. 4, illustrates a set of possible solutions using our approach, according to the same number of nodes and problem configuration.

Table 2
Computational result of TSP Problem.

Nodes	GA [43]	ACS [44]	AS [44]	FACO [44]
att48.tsp	34733.84	–	–	–
berlin52.tsp	8866.87	–	–	–
St.70	–	697.8	704.1	695
eil76.tsp	562.93	551.2	567.4	558

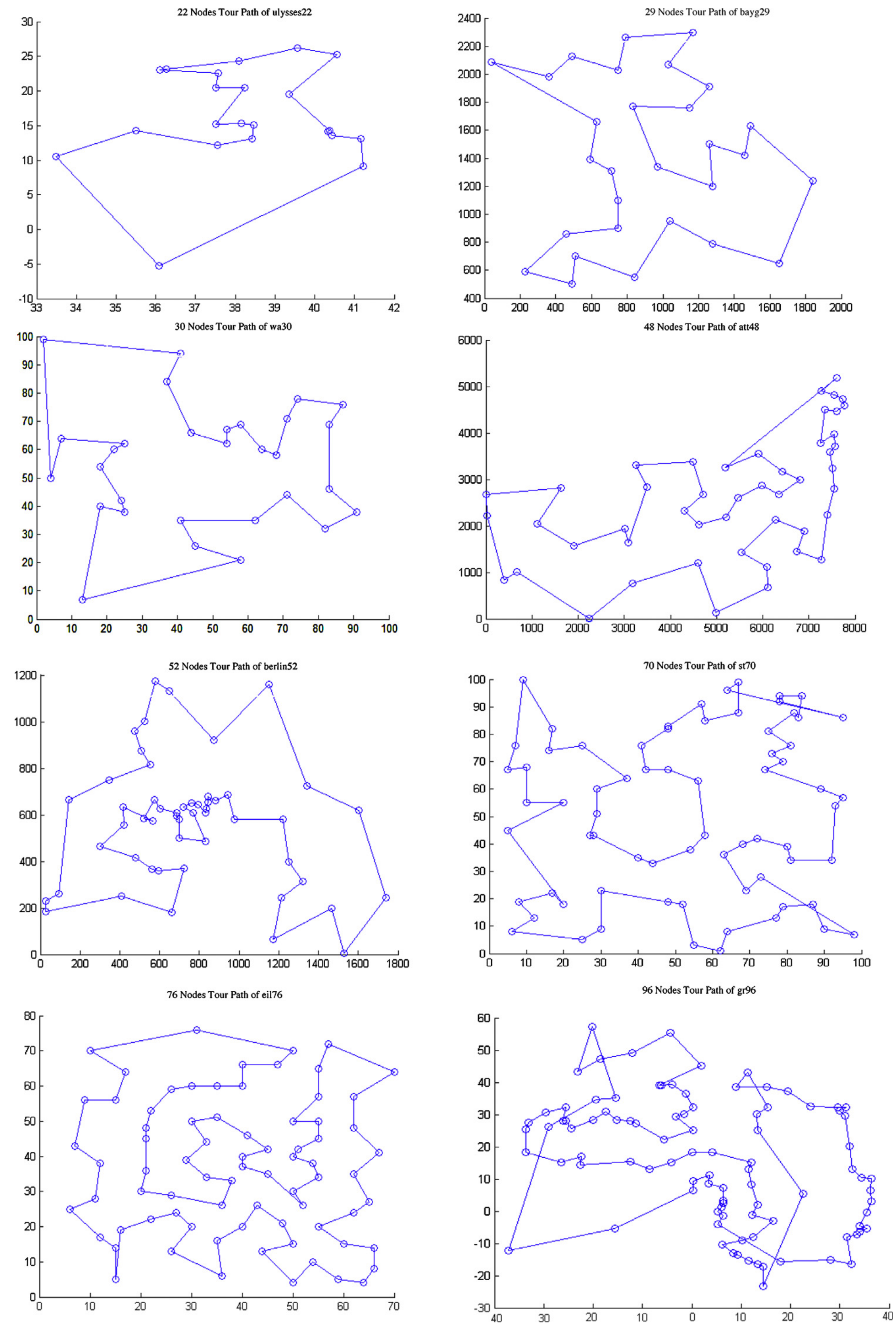

Fig. 4. TSP solutions for several nodes numbers.

6. Conclusions

We have seen many ways in which the individuals interact according to simple local rules that can also produce complex and adaptive social patterns of behaviors. Physical constraints limit certain kinds of coordinated behaviors, in particular those that require collision avoidance, and in many systems it is the adaptation to physical constraints that results in the interesting behavior. For instance, flocking and schooling, those remarkable group dynamism, result when a repulsive force is added to the force that attracts individuals toward the center of the aggregation.

This paper presented a novel hybrid method combining the PSO and ACO algorithms. The performance is validated using the TSP benchmark and the empirical results considering the completion time and the best length, illustrated that the proposed method is efficient.

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.

We would like to extend our thanks to Miss Mabrouk Rawia from the High Institute of Industrial Systems of Gabes-Tunisia, for her support.

References

[1] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence-from Natural to Artificial Systems, Oxford University Press, Oxford, 1999.
[2] S. Camazine, J.L. Deneubourg, N.R. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau, Self Organization in Biological Systems, Princeton University Press, Princeton, 2001.
[3] M. Burd, Ecological consequences of traffic organisation in ant societies, Physica A: Stat. Theor. Phys. 372 (1) (2006) 124–131.
[4] A. Dussutour, V. Fourcassi, D. Helbing, J.L. Deneubourg, Optimal traffic organization in ants under crowded conditions, Nature 428 (2004) 70–73.
[5] K. Vittori, G. Talbot, J. Gautrais, V. Fourcassi, A.F. Araujo, G. Theraulaz, Path efficiency of ant foraging trails in an artificial network, J. Theor. Biol. 239 (2006) 507–515.
[6] S.N. Beshers, J.H. Fewell, Models of division of labor in social insects, Annu. Rev. Entomol. 46 (1) (2001) 413–440.
[7] C. Detrain, J.L. Deneubourg, Self-organized structures in a superorganism: do ants "behave" like molecules, Phys. Life Rev. 3 (3) (2006) 162–187.
[8] E. Bonabeau, G. Theraulaz, J.L. Deneubourg, S. Aron, S. Camazine, Self-organization in social insects, Trends Ecol. Evol. 12 (5) (1997) 188–193.
[9] G.K. Purushothama, L. Jenkins, Simulated annealing with local search: a hybrid algorithm for unit commitment, IEEE Trans. Power Syst. 18 (1) (2003) 273–278.
[10] J. Kennedy, R.C. Eberhart, Particle swarm optimization, IEEE Int. Conf. Neural Netw. (1995) 1942–1948.
[11] V. Miranda, Evolutionary algorithms with particle swarm movements., in: In IEEE 13th International Conference on Intelligent Systems Application to Power Systems, November, 2005, pp. 6–21.
[12] W. Elloumi, Adel M. Alimi, Combinatory optimization of ACO and PSO, in: International Conference on Metaheuristique and Nature Inspired Computing, October, 2008, pp. 1–8.
[13] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents., IEEE Trans. Syst. Man Cybern. B Cybern. 26 (February (2)) (1996) 29–41.
[14] T.G. Crainic, M. Grendeau, Coperative parallel tabu search for capacitated network design, J. Heuristics 8 (2002) 601–627.
[15] M. Nowostawski, R. Poli, Parallel genetic algorithms taxonomy, in: In Proc. 3rd International Conference on Knowledge-Based Intelligent Information Engineering Systems, 1999, pp. 88–92.
[16] E. Cantu-Paz, A Survey PF Parallel Genetic Algorithms Technical Report Illegal 97003, The University of Illinois, 1997.
[17] M. Middendorf, F. Reischle, H. Schmeck, Multi colony ant algorithms, J. Heuristics 8 (2002) 305–320.
[18] S. Baskar, P.N. Suganthan, A novel current particle swarm optimization, Proc. IEEE Congr. Evol. Comput. 1 (2004) 792–796.
[19] D.G. Cabrero, C. Armero, D.N. Ranasinghe, The travelling salesman's problem: a self-adapting PSO–ACS algorithm, in: International Conference on Industrial and Information Systems, 2007.
[20] E. Talbi, A taxonomy of hybrid metaheuristics, J. Heuristics 8 (2002) 541–564.
[21] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents., IEEE Trans. Syst. Man Cybern. (1996) 1–13.
[22] A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies, in: I. n. F. Varela, P. Bourgine (Eds.), Proceedings of ECAL'91 – First European Conference on Artificial Life, Elsevier Publishing, Paris, France, 1992, pp. 134–142.
[23] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the travelling salesman problem, IEEE Trans. Evol. Comput. (1997).
[24] J. Kennedy, R. Mendez, Population structure and particle swarm performance, in: Proc. IEEE. Congress on Coevolutionary Computation, vol. 2, 2002, pp. 1671–1676.
[25] R.C. Eberhart, P. Simpson, R. Dobbins, Computational intelligence, in: PC Tools: Academic Chapter 6, 1996, pp. 212–226.
[26] Y. Yang, M. Kamel, Clustering ensemble using swarm intelligence, in: Proc. IEEE Swarm Intelligence Symposium, 2003, pp. 65–71.
[27] W. Elloumi, N. Rokbani, Adel M. Alimi, Ant Supervised By PSO, in: International Symposium on Computational Intelligence and Intelligent Informatics, October, 2009, pp. 161–166.
[28] W. Elloumi, Adel M. Alimi, A More Efficient MOPSO for Optimization, in: The eight ACS/IEEE International Conference on Computer Systems and Applications, May, 2010, pp. 1–7.
[29] Z. Michalewicz, D.B. Fogel, How to Solve It: Modern Heuristics, Springer, Berlin, New York, 2004.
[30] D.T. Pham, D. Karaboga, Intelligent Optimisation Techniques, Springer, London, New York, 2000.
[31] J.M. Zurada, Introduction to Artificial Neural Systems, West, St. Paul, 1992.
[32] V. Ramos, C. Fernandes, A.C. Rosa, Social cognitive maps, swarm collective perception and distributed search on dynamic landscapes, J. New Media Neural Cogn. Sci., NRW, Germany (2005), for his inaugural issue.
[33] S. Garnier, J. Gautrais, G. Theraulaz, The biological principles of swarm intelligence, in: Swarm Intelligence, 2007.
[34] A.J. Ouyang, Y.Q. Zhou, An improved PSO–ACO algorithm for solving large-scale TSP, J. Adv. Mater. Res. 1 (2010) 1154–1158.
[35] S. Nonsiri, S. Supratid, Modifying ant colony optimization, in: IEEE Conference on Soft Computing in Industrial Applications, 2008.
[36] F. Van den Bergh, A.P. Engelbrecht, A co-operative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 225–239.
[37] W. Elloumi, N. Baklouti, A. Abraham, Adel 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.
[38] K. Xu, M. Jiang, D. Yuan, Parallel artificial bee colony algorithm for the traveling salesman problem, in: Proceedings of the 2nd International Conference on Computer and Information Application (ICCIA 2012), 2012, pp. 663–667.
[39] L.P. Wong, M. Low, C.S. Chong, Bee colony optimization with local search for traveling salesman problem, in: Proceedings of the 6th IEEE International Conference on Industrial Informatics (INDIN 2008), 2008, pp. 1019–1025.
[40] S. Liu, A powerful genetic algorithm for traveling salesman problem, in: Proceedings of the course Principles of Artificial Intelligence, 2012.
[41] Z.H. Ahmed, Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator, Int. J. Biometrics & Bioinform. 3 (6) (2010) 96–105.
[42] W. Elloumi, N. Baklouti, A. Abraham, Adel M. Alimi, The multi-objective hybridization of particle swarm optimization and fuzzy ant colony optimization, J. Intell. Fuzzy Syst. (2014), http://dx.doi.org/10.3233/IFS-131020, Netherlands.
[43] J.Q. Yang, J.G. Yang, A novel genetic algorithm for traveling salesman problem based on neighborhood code, in: Second International Conference on Intelligent Networks and Intelligent Systems, 2009, pp. 429–432.
[44] Y.H. Chang, C.W. Chang, C.W. Tao, H.W. Lin, J.S. Taur, Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization, Expert Syst. Appl. 39 (2012) 3624–3633.