Revisiting Elitism in Ant Colony Optimization

Tony White, Simon Kaegi, Terri Oda

School of Computer Science, Carleton University
1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6
arpwhite@scs.carleton.ca, skaegi@sympatico.ca, terri@zone12.com

Abstract. Ant Colony Optimization (ACO) has been applied successfully in solving the Traveling Salesman Problem. Marco Dorigo et al. used Ant System (AS) to explore the Symmetric Traveling Salesman Problem and found that the use of a small number of elitist ants can improve algorithm performance. The elitist ants take advantage of global knowledge of the best tour found to date and reinforce this tour with pheromone in order to focus future searches more effectively. This paper discusses an alternative approach where only local information is used to reinforce good tours thereby enhancing the ability of the algorithm for multiprocessor or actual network implementation. In the model proposed, the ants are endowed with a memory of their best tour to date. The ants then reinforce this "local best tour" with pheromone during an iteration to mimic the search focusing of the elitist ants. The environment used to simulate this model is described and compared with Ant System.

Keywords: Heuristic Search, Ant Algorithm, Ant Colony Optimization, Ant System, Traveling Salesman Problem.

1 Introduction

Ant algorithms (also known as Ant Colony Optimization) are a class of heuristic search algorithms that have been successfully applied to solving NP hard problems [1]. Ant algorithms are biologically inspired from the behavior of colonies of real ants, and in particular how they forage for food. One of the main ideas behind this approach is that the ants can communicate with one another through indirect means by making modifications to the concentration of highly volatile chemicals called pheromones in their immediate environment.

The Traveling Salesman Problem (TSP) is an NP complete problem addressed by the optimization community having been the target of considerable research [7]. The TSP is recognized as an easily understood, hard optimization problem of finding the shortest circuit of a set of cities starting from one city, visiting each other city exactly once, and returning to the start city again. Formally, the TSP is the problem of finding the shortest Hamiltonian circuit of a set of nodes. There are two classes of TSP problem: symmetric TSP, and asymmetric TSP (ATSP). The difference between the two classes is that with symmetric TSP the distance between two cities is the same regardless of the direction you travel, with ATSP this is not necessarily the case.

Ant Colony Optimization has been successfully applied to both classes of TSP with good, and often excellent, results. The ACO algorithm skeleton for TSP is as follows [7]:

```
procedure ACO algorithm for TSPs
  Set parameters, initialize pheromone trails
  while (termination condition not met) do
    ConstructSolutions
    ApplyLocalSearch % optional
    UpdateTrails
  end
end ACO algorithm for TSPs
```

The earliest implementation, Ant System, was applied to the symmetric TSP problem initially and as this paper presents a proposed improvement to Ant System this is where we will focus our efforts.

While the ant foraging behaviour on which the Ant System is based has no central control or global information on which to draw, the use of global best information in the Elitest form of the Ant System represents a significant departure from the purely distributed nature of ant-based foraging. Use of global information presents a significant barrier to *fully* distributed implementations of Ant System algorithms in a live network, for example. This observation motivates the development of a fully distributed algorithm – the Ant System Local Best Tour (AS-LBT) – described in this paper. As the results demonstrate, it also has the by-product of having superior performance when compared to the Elitest form of the Ant System (AS-E). It also has fewer defining parameters.

The remainder of this paper consists of 5 sections. The next section provides further detail for the algorithm shown above. The Ant System Local Best Tour (AS-LBT) algorithm is then introduced and the experimental setup for its evaluation described. An analysis section follows, and the paper concludes with an evaluation of the algorithm with proposals for future work.

2 Ant System (AS)

Ant System was the earliest implementation of Ant Colony Optimization meta heuristic. The implementation is built on top of the ACO algorithm skeleton shown above. A brief description of the algorithm follows. For a comprehensive description of the algorithm, see [1, 2, 3 or 7].

2.1 Algorithm

Expanding upon the algorithm above, an ACO consists of two main sections: *initialization* and a *main loop*. The main loop runs for a user-defined number of iterations. These are described below:

Initialization

- Any initial parameters are loaded.
- Each of the roads is set with an initial pheromone value.
- Each ant is individually placed on a random city.

Main loop begins

Construct Solution

- Each ant constructs a tour by successively applying the probabilistic choice function and randomly selecting a city it has not yet visited until each city has been visited exactly once.

$$p_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{l \in N_i^k} \left[\tau_{il}(t)\right]^{\alpha} \cdot \left[\eta_{il}\right]^{\beta}}$$

- The probabilistic function, $p_{ij}^{k}(t)$, is designed to favor the selection of a road that has a high pheromone value, τ , and high visibility value, η, which is given by: $1/d_{ij}$, where d_{ij} is the distance to the city. The pheromone scaling factor, α , and visibility scaling factor, β , are parameters used to tune the relative importance of pheromone and road length in selecting the next city.

Apply Local Search

- Not used in Ant System, but is used in several variations of the TSP problem where 2-opt or 3-opt local optimizers [7] are used.

Best Tour check

- For each ant, calculate the length of the ant's tour and compare to the best tour's length. If there is an improvement, update it.

Update Trails

- Evaporate a fixed proportion of the pheromone on each road.
- For each ant perform the "ant-cycle" pheromone update.
- Reinforce the best tour with a set number of "elitist ants" performing the "ant-cycle" pheromone update.

In the original investigation of Ant System algorithms, there were three versions of Ant System that differed in how and when they laid pheromone. The "Ant-density" heuristic updates the pheromone on a road traveled with a fixed amount after every step. The "Ant-quantity" heuristic updates the pheromone on a road traveled with an amount proportional to the inverse of the length of the road after every step. Finally, the "Ant-cycle" heuristic first completes the tour and then updates each road used with an amount proportional to the inverse of the total length of the tour.

Of the three approaches "Ant-cycle" was found to produce the best results and subsequently receives the most attention. It will be used for the remainder of this paper.

2.2 Discussion

Ant System in general has been identified as having several good properties related to directed exploration of the problem space without getting trapped in local minima [1]. The initial form of AS did not make use of elitist ants and did not direct the search as well as it might. This observation was confirmed in our experimentation performed as a control and used to verify the correctness of our implementation.

The addition of elitist ants was found to improve ant capabilities for finding better tours in fewer iterations of the algorithm, by highlighting the best tour. However, by using elitist ants to reinforce the best tour the problem now takes advantage of global data with the additional problem of deciding on how many elitist ants to use. If too many elitist ants are used the algorithm can easily become trapped in local minima [1, 3]. This represents the dilemma of exploitation versus exploration that is present in most optimization algorithms.

There have been a number of improvements to the original Ant System algorithm. They have focused on two main areas of improvement [7]. First, they more strongly exploit the globally best solution found. Second, they make use of a fast local search algorithm like 2-opt, 3-opt, or the Lin-Kernighan heuristic to improve the solutions found by the ants.

The algorithm improvements to Ant System have produced some of the highest quality solutions when applied to the TSP and other NP complete (or NP hard) problems [1].

As described in section 2.1, augmenting AS with a local search facility would be straightforward; however, it is not considered here. The area of improvement proposed in this paper is to explore an alternative to using the globally best tour (GBT) to reinforce and focus on good areas of the search space. The Ant System Local Best Tour algorithm is described in the next section.

3 Ant System Local Best Tour (AS-LBT)

The use of an elitist ant in Ant System exposes the need for a global observer to watch over the problem and identify what the best tour found to date is on a per iteration basis. As such, it represents a significant departure from the purely distributed AS algorithm.

The idea behind the design of AS-LBT is specifically to remove this notion of a global observer from the problem. Instead, each individual ant keeps track of the best tour it has found to date and uses it in place of the elitist ant tour to reinforce tour goodness.

It is as if the scale of the problem has been brought down to the ant level and each ant is running its individual copy of the Ant System algorithm using a single elitist ant. Remarkably, the ants work together effectively even if indirectly and the net effect is very similar to that of using the pheromone search focusing of the elitist ant approach. In fact, AS-E and AS-LBT can be thought of as extreme forms of a Particle Swarm algorithm. In Particle Swarm Optimization (PSO), particles (effectively equivalent to ants in ACO) have their search process moderated by both local and global best solutions.

3.1 Algorithm

The algorithm used is identical to that described for Ant System with the replacement of the elitist ant step with the ant's local best tour step. Referring, once again, to the algorithm described in section 2.1, the following changes are made:
That is, where the elitist ant step was:

- Reinforce the best tour with a set number of "elitist ants" performing the "ant-cycle" pheromone update.

For Local Best Tour we now do the following:

- For each ant perform the "ant-cycle" pheromone update using its local best tour.

The rest of the Ant System algorithm is unchanged, including the newly explored tour's "ant-cycle" pheromone update.

3.2 Experimentation and Results

For the purposes of demonstrating AS-LBT we constructed an Ant System simulation and applied it to a series of TSP Problems from the TSPLIB95 collection [6]. Three symmetric TSP problems were studied: eil51, eil76 and kro101. The eil51 problem is a 51-city TSP instance set up in a 2 dimensional Euclidean plane for which the optimal tour is known. The weight assigned to each road comes from the linear distance separating each pair of cities. The problems eil76 and kro101 represent symmetric TSP problems of 76 and 101 cities respectively.

The simulation created for this paper was able to emulate the behavior of the original Ant System (AS), Ant System with elitist ants (AS-E), and finally Ant System using the local best tour (AS-LBT) approach described in section 2.

3.2.1 Parameters and Settings

Ant System requires you to make a number of parameter selections. These parameters are:

Pheromone sensitivity (α) = 1	Pheromone additive constant
Visibility sensitivity (β) = 5	Number of ants
Pheromone decay rate (ρ) = 0.5	Number of elitist ants
Initial pheromone (τ_0) = 10^{-6}	

In his original work on Ant System Marco Dorigo performed considerable experimentation to tune and find appropriate values for a number of these parameters

[3]. The values Dorigo found that provide for the best performance when averaged over the problems he studied were used in our experiments. These best-practice values are shown in the list above.

For those parameters that depend on the size of the problem our simulation made an effort to select good values based on knowledge of the problem and number of cities. Recent work [5] on improved algorithm parameters was unavailable to us when developing the LBT algorithm. We intend to explore the performance of the new parameters settings and will report the results in a future communication.

The Pheromone additive constant (Q) was eliminated altogether as a parameter by replacing it with the global best tour (GBT) length in the case of standard Ant System and the local best tour (LBT) length for the approach in this paper. We justify this decision by noting that Dorigo found that differences in the value of Q only weakly affected the performance of the algorithm and a value within an order of magnitude of the optimal tour length was acceptable. This means that the pheromone addition on an edge becomes:

$$\frac{L_{best}}{L_{ant}}$$ For a normal "ant-cycle" pheromone update

$$\frac{L_{best}}{L_{best}} = 1$$ For an elitist or LBT "ant-cycle" pheromone update

The key factor in the pheromone update is that it remains inversely proportional to the length of the tour and this still holds with our approach. The ants now are not tied to a particular value of Q in the event of a change in the number of cities in the problem. We consider the removal of a user-defined parameter another attractive feature of the LBT algorithm and a contribution of the research reported here.

For the number of ants, we set this equal to the number of cities, as this seems to be a reasonable selection according to the current literature [1, 3, 7].

For the number of elitist ants we tried various values dependent on the size of the problem and used a value of $1/6^{th}$ of the number of cities for the results reported in this paper. This value worked well for the relatively low number of cities we used in our experimentation but for larger problems this value might need to be tuned, possibly using the techniques used in [5]. The current literature is unclear on the best value of the number of elitest ants to be used.

With AS-LBT, all ants perform the LBT "ant-cycle" update so subsequently the number of elitist ants is not needed. We consider the removal of the requirement to specify a value for the number of elitest ants an advantage. Hereafter, we refer to AS with elitest ants as AS-E.

3.2.2 Results

Using the parameters from the previous section, we performed 100 experiments for eil51, eil76 and kro101; the results are shown in Figures 1, 2 and 3 respectively. In the case of eil51 and eil76, 2000 iterations of each algorithm were performed, whereas 3500 iterations were used for kro101. The results of the experimentation showed considerable promise for AS-LBT. While experiments for basic AS were performed, they are not reported in detail here as they were simply undertaken in order to validate the code written for AS-E and AS-LBT.

Figure 1. Difference between LBT and Elitest Algorithms (eil51)

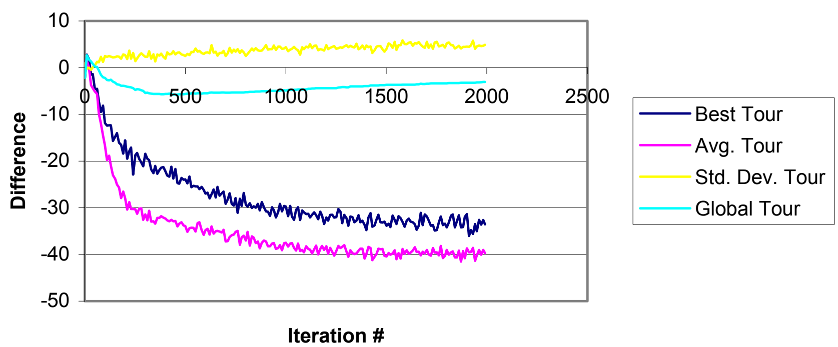
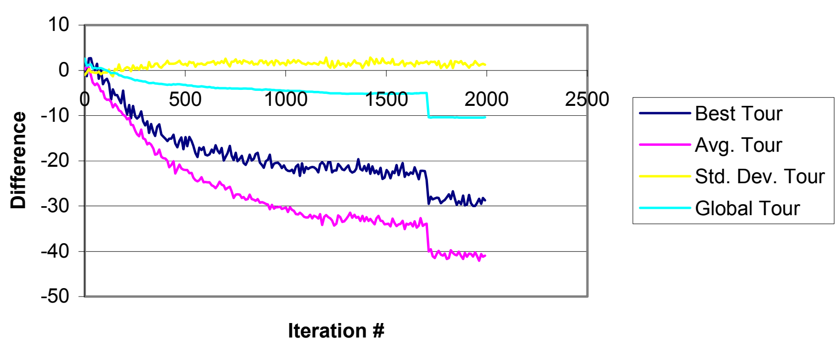
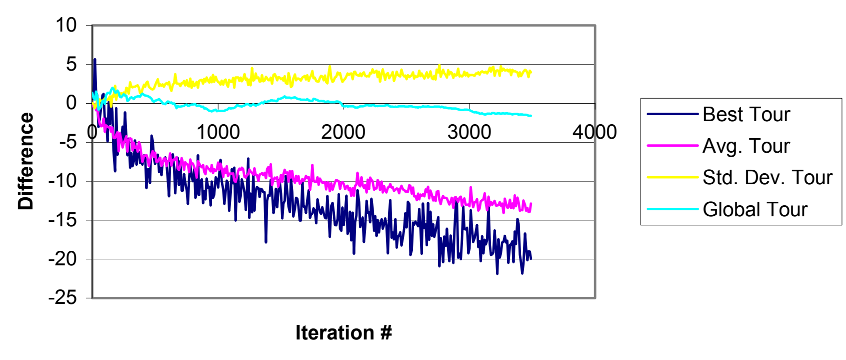


Figure 2. Difference between LBT and Elitest Algorithms (eil76)



Figures 1, 2 and 3, each containing 4 curves, require some explanation. Each curve in each figure is the difference between the AS-LBT and AS-E per-iteration average of the 100 experiments performed. Specifically, the “Best Tour” curve represents the difference in the average best tour per iteration between AS-LBT and AS-E. The “Avg. Tour” curve represents the difference in the average tour per iteration between AS-LBT and AS-E. The “Std. Dev. Tour” curve represents the difference in the standard deviation of all tours per iteration between AS-LBT and AS-E. Finally, the “Global Tour” curve represents the difference in the best tour found per iteration between AS-LBT and AS-E. As the TSP is a minimization problem, negative difference values indicate superior performance for AS-LBT. The

Figure 3. Difference between LBT and Elitest Algorithms (kro101)



most important measure is the "Global Tour" measure, at least at the end of the experiment. This information is summarized in Table 1, below.

Table 1. Difference in Results for AS-LBT and AS-E

	Best Tour	Average Tour	Std. Dev Tour	Global Tour
eil51	-33.56	-39.74	4.91	-3.00
eil76	-29.65	-41.25	1.08	-10.48
Kro101	-19.97	-12.86	3.99	-1.58

The results in Table 1 clearly indicate the superior nature of the AS-LBT algorithm. The "Global Tour" is superior, on average, in all 3 TSP problems at the end of the experiment. The difference between AS-E and AS-LBT is significant for all 3 problems for a t-test with an α value of 0.05. Similarly, the "Best Tour" and "Average Tour" are also better, on average, for AS-LBT. The results for eil76 are particularly impressive, owing much of their success to the ability of AS-LBT to find superior solutions at approximately 1710 iterations.

The one statistic that is higher for AS-LBT is the average standard deviation of tour length on a per-iteration basis. This, too, is an advantage for the algorithm in that it means that there is still considerable diversity in the population of tours being explored. It is, therefore, more effective at avoiding local optima.

4 Analysis

Best Tour Analysis: As has been shown in the Results section, AS-LBT is superior to the AS-E approach as measured by the best tour found. In this section we take a comparative look at the evolution of the best tour in all three systems and then a look at the evolution of the best tour found per iteration.

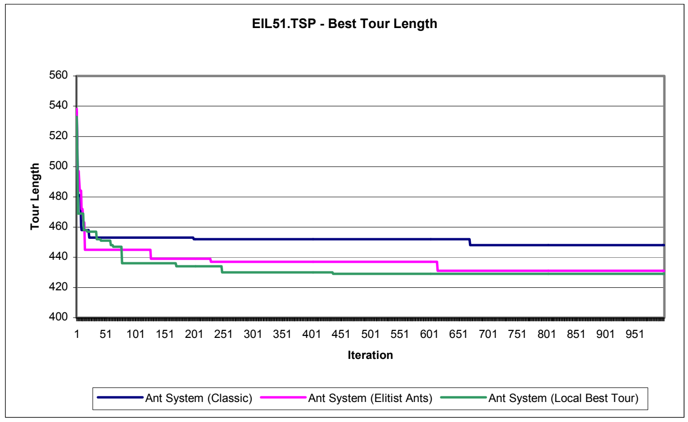


Figure 4. Evolution of Best Tour Length

In Figure 4, which represents a single typical experiment, we can see the key difference between AS-E and AS-LBT. Whereas AS-E quickly finds a few good results, holds steady and then improves in relatively large pronounced steps, AS-LBT improves more gradually at the beginning but continues its downward movement at a steadier rate. In fact, if one looks closely at the graph one can see that even the classical AS system has found a better result during the early stages of the simulation when compared to AS-LBT. However, by about iteration 75, AS-LBT has overtaken the other two approaches and continues to gradually make improvements and maintains its overall improvement until the end of the experiment. This is confirmed in Figure 1, which is the average performance of AS-LBT for eil51 over 100 experiments.

Overall, the behavior of AS-LBT could be described as slower but steadier. It takes slightly longer at the beginning to focus pheromone on good tours but after it has, it improves more frequently and steadily and on average will overtake the other two approaches given enough time. Clearly this hypothesis is supported by experimentation with the eil76 and kro101 TSP problem datasets as shown in Figures 2 and 3.

***Average Tour Analysis**:* In the Best Tour Analysis we saw that there was a tendency for the AS-LBT algorithm to gradually improve in many small steps. With our analysis of the average tour we want to confirm that the relatively high deviation of ant algorithms is working in the average case meaning that we are continuing to explore the problem space effectively. In this section we look at the average tour length per iteration to see if we can identify any behavioural trends.

In Figure 5 we see a very similar situation to that of the Best Tour Length per Iteration. The AS-LBT algorithm is on average exploring much closer to the optimal solution. Perhaps more importantly, the AS-LBT graph trend line is behaving very similarly in terms of its deviation as that with the other two systems. This suggests that the AS-LBT system is working as expected and is in fact searching in a better-focused fashion closer to the optimal solution.

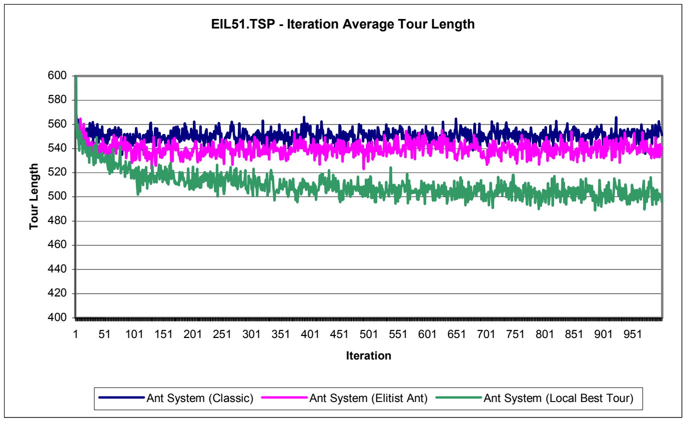


Figure 5. Average Tour Length for Individual Iterations

Evolution of the Local Best Tour: The Local Best Tour approach is certainly very similar to the notion of elitist ants; only it is applied at the local level instead of at the global level. In this section we look at the evolution of the local best tour in terms of the average and worst tours, and compare them with the global best tour used by elitist ants.

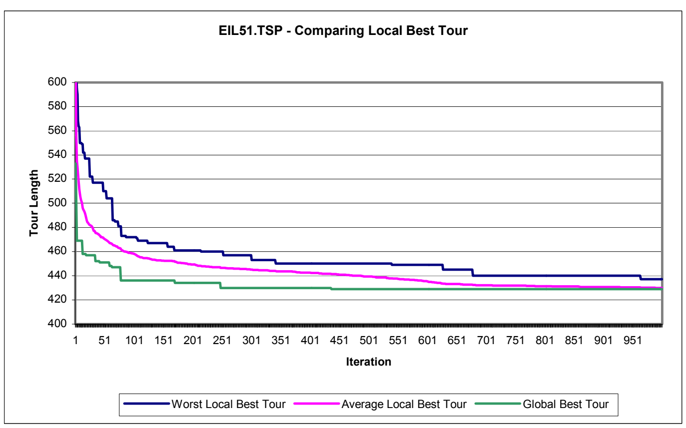


Figure 6. Evolution of the Local Best Tour

From Figure 6 we can see that over time both the average and worst LBTs approach the value of global best tour. In fact the average in this simulation is virtually the same as the global best tour. From this figure, it is clear that the longer the simulation runs the closer the LBT “ant-cycle” pheromone update becomes to that of an elitist ant’s update scheme.

5 Discussion and Future Work

Through the results and analysis shown in this paper, Local Best Tour has proven to be an effective alternative to the use of the globally best tour for focusing ant search through pheromone reinforcement. In particular, the results show that AS-LBT has excellent average performance characteristics. By removing the need for the global information required for AS-E, we have improved the ease with which a parallel or live network implementation can be achieved; i.e. a completely distributed implementation of the TSP is possible.

Analysis of the best tour construction process shows that AS-LBT, while initially converging more slowly than AS-E, is very consistent at incrementally building a better tour and on average will overtake the AS-E approach early in the search of the problem space.

Average and best iteration tour analysis has shown that AS-LBT shares the same variability characteristics of the original Ant System that make it resistant to getting stuck in local minima. Furthermore, AS-LBT is very effective in focusing its search towards the optimal solution.

Finally, AS-LBT follows in the notion that the use of best tours to better focus an ant’s search is an effect optimization. The emergent behaviour of a set of autonomous LBT ants is to, in effect, become elitist ants over time.

As described earlier in this paper, a relatively straightforward way to further improve the performance of AS-LBT would be to add a fast local search algorithm like 2-opt, 3-opt or the Lin Kernighan heuristic. Alternatively, the integration of recent network transformation algorithms [4] should prove useful as local search operators.

Finally, future work should include the application of the LBT algorithm to other problems such as: the asymmetric TSP, the Quadratic Assignment Problem (QAP), the Vehicle Routing Problem (VRP) and other problems to which ACO has been applied [1].

6 Conclusions

This paper has demonstrated that an ACO algorithm using only local information can be applied to the TSP. The AS-LBT algorithm is truly distributed and is characterized by fewer parameters when compared to AS-E. Considerable experimentation has demonstrated that *significant* improvements are possible for 3 TSP problems. We believe that AS-LBT with the improvements outlined in the previous section will further enhance our confidence in the hypothesis and look forward to reporting on these improvements in a future research paper. Finally, we believe that a Particle Swarm Optimization algorithm, where search is guided by both local best tour and global best tour terms may yield further improvements in performance for ACO algorithms.

References

1. Bonabeau E., Dorigo M., and Theraulaz G. Swarm Intelligence From Natural to Artificial Systems. Oxford University Press, New York NY, 1999.
2. Dorigo M. and L.M. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1(1):53-66, 1997.
3. Dorigo M., V. Maniezzo and A. Colorni. The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41, 1996.
4. Dumitrescu A. and Mitchell J., Approximation Algorithms for Geometric Optimization Problems, in the Proceedings of the 9th Canadian Conference on Computational Geometry, Queen's University, Kingston, Canada, August 11-14, 1997, pp. 229-232.
5. Pilat M. and White T., Using Genetic Algorithms to optimize ACS-TSP. In Proceedings of the 3rd International Workshop on Ant Algorithms, Brussels, Belgium, September 12-14 2002.
6. Reinelt G. TSPLIB, A Traveling Salesman Problem Library. ORSA Journal on Computing, 3:376-384, 1991.
7. Stützle T. and Dorigo M. ACO Algorithms for the Traveling Salesman Problem. In K. Miettinen, M. Makela, P. Neittaanmaki, J. Periaux, editors, Evolutionary Algorithms in Engineering and Computer Science, Wiley, 1999.