A Local Pheromone Initialization Approach for Ant Colony Optimization Algorithm

Abdelghani Bellaachia
School of Engineering and Applied Sciences
The George Washington University
Washington, DC, USA
bell@gwu.edu

Deema Alathel
School of Engineering and Applied Sciences
The George Washington University
Washington, DC, USA
atheld@gwmail.gwu.edu

Abstract—**Ant Colony Optimization Algorithms are the most successful and widely accepted algorithmic techniques based on the decentralized collaborative behavior of real ants when foraging for food. The initialization of pheromone in these algorithms is an important step because it dictates the speed of the system's convergence to the optimal solution. All the proposed initialization techniques in the literature use a single value to initialize the pheromone on all edges. In our paper, instead of using a constant or a pre-calculated value to initialize the pheromone the edges, we propose a local pheromone initialization technique that involves the ants initializing the edges, using local information, as they encounter the edges for the first time. We tested our proposed local initialization using the Ant Colony System algorithm to solve the Travelling Salesman Problem. Our approach, when compared to the standard initialization approaches, provided better results in more than 70% of the tested datasets. Also, our algorithm did not require an initialization for all edges. In general, our local pheromone initialization approach was successful in achieving a balance between the solution quality and the time required to construct that solution even in the cases in which it was not able to find the optimal path.**

Keywords- Ant colony system; Ant colony optimization; Artificial agents; Bio-inspired algorithms; Travelling salesman problem.

I. INTRODUCTION

Most, if not all, researchers working with ACO algorithms and their applications focus on the details of the solution construction, probability calculations, and finding new approaches for improving the way ants communicate and process information. However, pheromone initialization has been neglected where researchers opt to use a constant value for the initialization, whether it has been pre-calculated or estimated. But properly initializing the pheromone level affects the speed of the system's convergence to the optimal solution, and we believe that it deserves more attention than it has been given so far. In this paper, we propose a new approach for initializing the pheromone level on edges locally and we show through conducted experiments the advantages of adopting such an approach. In our algorithm, we treat the process of pheromone initialization as one of the tasks assigned to ants rather than treating it as a preprocessing step. In the remainder of this paper we define our local initialization technique, present the experiments that were carried as part of our work, and compare the results to some standard applications of the Ant Colony System (ACS) algorithm to the travelling salesman problem (TSP) using several datasets of varying sizes.

II. RELATED BACKGROUND

A. Ant Colony Obtimization Algorithms

ACO algorithms is a family of techniques that falls under swarm intelligence algorithms. ACO was first proposed by Dorigo in 1992 [4]. ACO is a probabilistic technique for solving optimization problems by mimicking the foraging behavior of ants when they are searching within their colony for a good food source [6][2][3][13]. In the context of the travelling salesman problem (TSP) [8][14][15][16], we can think of the first city where the salesman starts his tour as the ants' nest. In such a scenario, artificial ants are dispensed from the source city into the network to imitate the foraging behavior of real ants in search for a good food source, which in our case is the shortest tour passing through all cities. Just like real ants, the artificial ants search the solution space and update the pheromone level on traversed paths, increasing the pheromone level on paths leading to *good* solutions. The pheromone update process increases the probability of subsequent ants choosing the *good* paths that lead to the shorter tours and decreases their probability of choosing the other paths. In this manner, the system moves from an unstable stage, where no path is necessarily better than the other, to a stable one where certain paths emerge as being the best ones leading to the *best* food sources (shortest tours). To realistically mimic the behavior of real ants, the artificial ants are dispatched to search for the solution over several iterations in the system; these iterations lead to the system's convergence to the optimal solution [7]. At each step within an iteration, an ant k positioned at node x calculates the probability of moving to each neighboring node y and eventually picks the edge connecting to the node that yielded the highest probability. In general, the probability p^k_{xy} is calculated as:

$$p^k_{xy} = \frac{(\tau_{xy})^\alpha (\eta_{xy})^\beta}{\sum_{z \in N^k_x} (\tau_{xz})^\alpha (\eta_{xz})^\beta} \tag{1}$$

where τ_{xy} is the pheromone level on the edge xy, η_{xy} is the desirability of the move from node x to node y, α and β are parameters that control the influence of τ_{xy} and η_{xy} respectively,and N^k_x refers to ant k's neighborhood of *unvisited*

978-1-4799-2030-3 /14/$31.00 ©2014 IEEE

TABLE I. SUGGESTED PARAMETER SETTINGS FOR ACO ALGORITHMS WHEN APPLIED TO TSP

ACO Algorithm	α	β	ρ	k	τ_0
AS	1	2 to 5	0.5	n	k/C_{nn}
EAS	1	2 to 5	0.5	n	$(n + k)/\rho C_{nn}$
MMAS	1	2 to 5	0.02	n	$1/\rho C_{nn}$
ACS	1	2 to 5	0.1	10	$1/nC_{nn}$

nodes when positioned at node x. It is noteworthy to note that in ACO algorithms in general, η_{xy} is considered as an a priori *desirability* computed by heuristics while τ_{xy} is a posteriori indication about the *goodness* of the move. Each ACO algorithm uses a different variation of Equation 1.

B. Importance of Initial Pheromone Level

Typically, an ideal value for the initial pheromone level on edges τ_0 in ACO algorithms would be one that is as close as possible to the average pheromone level expected to be deposited by an ant on an edge during one iteration [6]. Choosing a very small value will slow the convergence process and may result in the system not getting a chance to reach the optimal solution. Initializing the pheromone level using a large value will cause a fast convergence, which means the system will not take its time to explore other possible paths and eventually it will be stuck at a suboptimal solution. Also the choice between whether to use a constant value for τ_0 or to pre-calculate it affects the quality of the best solution reached.

In ACO algorithms in general, the initial pheromone level is usually initialized either using a carefully chosen constant or using a pre-calculated value obtained from running a quick suboptimal path construction algorithm [11][5], such as the nearest-neighbor algorithm, on the network. In both cases, the pheromone level on all edges in the system is initialized using the same value τ_0. If a small constant value will be used for initializing τ_0, then it needs to be carefully chosen to avoid extreme convergence situations as previously discussed.

C. Initial Pheromone Level in Traditional ACO Algorithms for the Travelling Salesman Problem

There are different ways to pre-calculate a value to initialize τ_0 in ACO algorithms. Even for a certain problem, like the TSP, there is no standard approach to be followed. However, the standard application of each ACO algorithm for the TSP has a set of suggested parameters associated with each that have been proven to yield good results. For example, when the Ant System (AS) algorithm is applied to the TSP problem, Dorigo et al. [10] suggest initializing τ_0 as follows:

$$\tau_0 = k/C_{nn} \tag{2}$$

where k is the number of ants and C_{nn} is a pre-calculated value corresponding to the length of a tour constructed by applying the nearest neighbor algorithm [1]. In Elitist Ant System algorithm (EAS) [9][4] more parameters are considered in the initialization, such that:

$$\tau_0 = (n + k)/\rho C_{nn} \tag{3}$$

where n is the number of cities, and ρ is the evaporation coefficient used in the pheromone evaporation process. *MAX-MIN* AS (MMAS) [18][19][17] suggests an initialization that only considers the nearest neighbor tour length in addition to the pheromone evaporation coefficient:

$$\tau_0 = 1/\rho C_{nn} \tag{4}$$

Similarly, the Ant Colony System algorithm (ACS) uses the number of cities instead of the evaporation coefficient [12], as follows:

$$\tau_0 = 1/nC_{nn} \tag{5}$$

Table 1 summarizes the suggested parameter values used by these algorithms in general.

III. PROPOSED PHEROMONE INITIALIZATION APPROACH

In this section we present our local pheromone initialization approach. Our technique considers only accessible local information that an artificial ant can use to generate an initial value for the uninitialized edges. Our local initialization methodology proved its ability to contribute to constructing better solution in most cases, and even in the cases where the solutions were suboptimal our approach compensates for that by greatly decreasing the number of iterations needed to construct the solution.

A. Rational behind Proposed Approach

Anyone reviewing the ant algorithms literature would notice that all of the proposed ACO algorithms (and their variations) focus on devising new ways for the agents to interact and probabilistically move on the graph while neglecting any attempts to improve the way the basic parameters are initialized. Although the suggested parameters summarized in Table 1 have been proven to provide good results, but that does not mean that there is no room for improvement. That was the motivation behind our curiosity that was ignited by the interest to investigate the possibility of applying a new pheromone initialization approach to further enhance the results.

Note that the suggested pheromone initialization techniques in the literature, regardless whether a constant value or a pre-calculated value is used, propose calculating a single value and using it to initialize the pheromone level on *all* edges as part of a preprocessing step, which can be considered as what we call a *global pheromone initialization* approach. However, we categorize our proposed technique as a *local pheromone initialization* approach since we propose calculating the initial pheromone levels locally within the current node's neighborhood rather than using a *one-size-fits-all* approach. In order for any ACO system to converge properly to the optimal solution in a timely manner, the initial pheromone level on edges should be a value close to what an ant is expected to deposit on an edge during a single iteration. The incentive behind our proposed approach is that since we are looking for a value that is close to what an ant would deposit, and since that ants determine the amount of pheromone to be deposited based on calculations made locally which yields different deposited amounts based on the quality of the path, then it does not make sense to initialize all edges using the same value especially when knowing that not all edges are of equal quality in the path construction process. Thus, we propose initializing each edge upon being encountered by an ant for the first time, before the ant proceeds with calculating p^k_{xy}.

B. Applying Local Pheromone Initialization Algorithm to Ant Colony System for the Travelling Salesman Problem

For our work, we chose to experiment with the pheromone initialization in the ACS algorithm when used to solve the TSP. Our choice was not arbitrary; AS is the first of the ACO algorithms to be proposed while ACS is an extension to it that has a better exploitation for the local search carried by ants and it performs pheromone evaporation and deposit on the best solutions found so far at the end of each iteration rather than updating all paths. MMAS uses an initial pheromone value that is a function of the pheromone evaporation coefficient ρ, which is a parameter that also needs to be properly controlled in the experiments. If MMAS was the choice for our experiments it would have been hard to determine whether the results obtained are attributed to the proposed initialization technique or to the value used for ρ. In EAS analyzing the results would be even more complicated since the initial value involves, in addition to ρ, the number of ants used.

Recall from Equation 5 that the initial pheromone level τ_0 in ACS for TSP is a function of the number of cities n in the problem, and the length of the tour constructed using the nearest neighbor algorithm C_{nn}.

In our local initialization approach, we delegate the pheromone initialization task to the ants as opposed to treating it as a preprocessing step. Each ant initializes the edges before deciding which one to cross (upon their first encounter). Our goal is threefold:

1. Use local information that ant k has access to when it is at node x, i.e. information in node x's surrounding neighborhood N^k_x, to determine the proper initial pheromone level on the neighboring edges to reduce the effect of external factors (from distant nodes).
2. Avoid initializing unused (un-encountered) edges.
3. Reduce the number of iterations needed (time spent) for the system to converge.

The basic idea behind our technique is to use local information from the surrounding neighborhood to initialize the pheromone level on each edge xy before being potentially crossed for the first time by using an initial local value τ^0_{xy} instead of initializing with τ_0. In our experiments, τ^0_{xy} is defined as the inverse of the sum of weights (distances) associated with edges $z \in N^k_x$ that can be potentially crossed by ant k from the current node (city) x:

$$\tau^0_{xy} = \frac{1}{\sum_{z \in N^k_x} d_{xz}} \quad (6)$$

where d_{xz} is the weight (distance) associated with an edge connecting node (city) x to node (city) z that belongs to node x's neighborhood N^k_x.

To further illustrate how our pheromone initialization approach is incorporated into the ACS algorithm, imagine the following scenario: an ant k is on node x and wants to determine which node y in its neighborhood N^k_x should it move to next. If an edge xy has not been encountered by any ant so far (i.e. has not been initialized), then it needs to be initialized before calculating the probability of the moves p^k_{xy}. Note that the term *encountered* is used rather than *crossed* because the local initialization step covers all potential uninitialized edges (regardless whether an edge gets crossed or not).

An ant k can face one of three possible cases:

- *Case 1:* None of the encountered edges have been initialized yet. In this case, the ant initializes all potential edges using our proposed local pheromone initialization technique as described by Equation 6.
- *Case 2*: All potential neighboring edges have associated pheromone level values (i.e. they have already been initialized). In such a case, the ACS algorithm proceeds with the calculation of the probability p^k_{xy} of moving to a neighboring node.
- *Case 3*: Some potential neighboring edges have not been initialized while the rest have pheromone values associated with them. When faced with a similar scenario, our approach only considers the potential uninitialized edges in the determination of the initial pheromone level when applying Equation 6 to initialize them.

Notice that not all edges will be initialized during the first iteration of the system; it is possible to have initializations in every iteration. It is expected however that the percentage of the initialized edges per iteration will decrease in the final iterations since that's when the system starts to converge to the optimal solution. Also note that the initialized edges will not necessarily have the same initial pheromone level; the initial pheromone level depends on the information (weights) within each neighborhood.

IV. Preliminary Experimental Evaluation

In this section we provide the details of the experiments that were conducted to validate the performance of our proposed local pheromone initialization algorithm. We compare our results to the ones obtained by using traditional pheromone initialization techniques. Namely we compare our algorithm to the basic pheromone initialization techniques that are used in ACS algorithms. We focus in the preliminary experiments on the different techniques' performances with respect to the quality of the best solution found, the time it took to reach the best solution (as the number of iterations required to converge to that solution).

A. Preliminary Experiments

In order to test the feasibility of our proposed local pheromone initialization scheme, a sample TSP dataset with 50 cities and randomly generated distances among them was used over 2500 iterations experimenting with different number of ants. The experiments were repeated 20 times to validate the results. For the sake of comparison, we compared running three

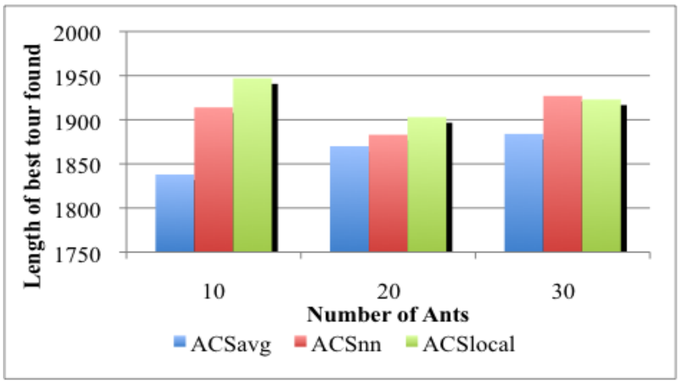

Figure 1. Comparing the length of the best tour found using the three different initialization techniques.

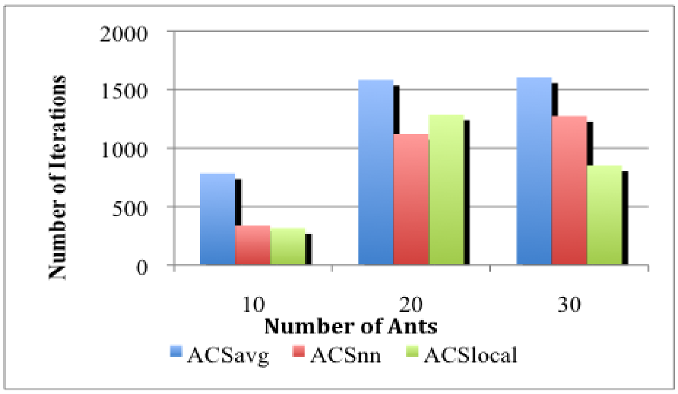

Figure 2. Comparing the number of iterations needed to construct the best tour found using the three different initialization techniques.

algorithms on the randomly generated dataset:

- ACS^{nn}: The ACS algorithm using the length of tour calculated using the nearest neighbor algorithm to initialize τ_0 (Equation 5).
- ACS^{avg}: The ACS algorithm using a pre-calculated constant value that initializes τ_0 using:

$$\tau_0 = 1/(n * avgDist) \tag{7}$$

  where n is the number of cities and *avgDist* refers to the average distance between any two cities in the used dataset.
- ACS^{local}: The ACS algorithm initializing τ_0 using the proposed local pheromone initialization technique described in Equation 6.

Note that we proposed the constant ACS^{avg} to serve as a common ground between the extreme approaches of ACS^{nn} and ACS^{local}.

B. Analysis of Preliminary Experiments

A quick glance at the results in Fig. 1 and Fig. 2 shows that ACS^{local} has a great potential for improving the performance of the ACS algorithm in solving the TSP. As we can see from examining the figures, our proposed local initialization algorithm achieved reasonable results when compared to ACS^{avg} and ACS^{nn} in general. Even though when 10 and 20 ants were used, ACS^{local} resulted in a best tour that had a length that is slightly worse that the ones reached by the other two algorithms, but we can see from Fig. 1 that ACS^{local}'s performance improves as the number of ants increases. In addition, when comparing Fig. 1 and Fig. 2 we even notice that for 10 and 20 ants, although the best tour length was longer for ACS^{local}, but on the other hand the algorithm caused a drastic decrease in the number of iterations needed to find the best tour as depicted in Fig. 2. A closer investigation of the results shows that, for example, when 10 ants were used, ACS^{local} increased the best tour length by about 50% when compared to the one resulted by ACS^{avg}, however our approach caused in the number of iterations greatly outweighs the increase in the length of the best tour found.

So, in general we can see that ACS^{local} performs better than the other typical techniques especially with respect to the number of iterations needed to find the best solution. As for the quality of the best solution, ACS^{local} does not always perform as well as ACS^{nn} or ACS^{avg} especially when using a fewer number of ants, but on the other hand it still provides a greater reduction in time which overshadows the drawback of the quality of the best solution found. Also, in general, we can see that increasing the number of ants contributes greatly to the improving the results obtained by using our proposed algorithm. As a result, and taking into consideration the size of the dataset and the limited number of rounds performed, the preliminary experiments have shown that the local pheromone initialization technique seems to be promising enough to be tested on a large-scale data.

V. Large-Scale Experimental Evaluation

Since our proposed local pheromone initialization technique showed some promise, we decided to perform additional experiments using large-scale datasets. Table 2 compares the results obtained by applying ACS^{nn} and ACS^{local} on 11 different datasets. The datasets provide a varying scope in size (the number next to the name of each dataset corresponds to the number of cities), which provides a way to monitor whether the dataset size affects the algorithms' performances. In all our experiments, we dispatched 50 ants over 50 iteration within each of 1000 runs.

Overall, ACS^{local} resulted in a better performance (or tied ACS^{nn}) with respect to finding shorter tours in 8 out of the 11 tested datasets (i.e. ~ 73% of the tested datasets), which confirms the validity of the results obtained in the preliminary experiments in the previous section. The dataset names highlighted in Table 2 indicate the cases in which ACS^{local} outperformed ACS^{nn}.

The results in general did not show an obvious trend or parameter that could have affected the outcomes. For example, both datasets *pr76* and *eil76* have 76 cities however ACS^{local} performed better in one while ACS^{nn} generated better results in the other. In the same manner, we noticed that the distances between the cities did not affect the results either.

TABLE II. SUMMARY OF EXPERIMENTAL RESULTS OBTAINED BY APPLYING THE ALGORITHMS TO 11 DIFFERENT DATASETS

Dataset	eil51		kroA100		att48		burmal14		bayg29		berlin52	
Algorithm	ACS^{nn}	ACS^{local}	ACS^{nn}	ACS^{local}	ACS^{nn}	ACS^{local}	ACS^{nn}	ACS^{local}	ACS^{nn}	ACS^{local}	ACS^{nn}	ACS^{local}
Best Tour Length	430	443	23122	**22862**	33274	**32564**	25	**25**	8681	**8445**	7311	7477
Worst Tour Length	551	601	28885	29484	41296	41594	32.14	32.14	10697	10831	9356	10135
Avg Tour Length	500	517	26047	26260	37153	**26994**	26.73	26.8	9704	**9695**	8320	8778
Ratio of Initialized Edges	100%	95%	100%	96%	100%	95%	100%	**91%**	100%	94%	100%	**92%**

Dataset	eil76		pr76		st70		ulysses16		ulysses22	
Algorithm	ACS^{nn}	ACS^{local}	ACS^{nn}	ACS^{local}	ACS^{nn}	ACS^{local}	ACS^{nn}	ACS^{local}	ACS^{nn}	ACS^{local}
Best Tour Length	556	597	116437	**114064**	696	**690**	51	**51**	54	**53**
Worst Tour Length	705	791	144605	156760	866	893	72	72	75	77
Avg Tour Length	639	690	131607	135816	780	788	57	**56**	59	60
Ratio of Initialized Edges	100%	94%	100%	**93%**	100%	95%	100%	**90%**	100%	**91%**

An important result that can be further analyzed is the percentage of edges that were not crossed and hence were not initialized in ACS^{local}. Such information may be useful for gaining further insight about the system and thus improving the overall performance. For instance note that in *burmal14*, *ulysses22*, and *pr76*, ACS^{local} provided better results in addition to initializing 91% of the edges for the two former datasets and 93% of the edges in the latter. Even when both algorithms perform equally well, such as in the case of the *burmal14* and *ulysses16* datasets, our algorithm still provided a better outcome since it resulted in the initialization of about 90% of the edges in both datasets. Passing such information to a data miner would certainly provide further understanding of the results.

We believe that ACSlocal's approach of utilizing the locally available information to ants to initialize the pheromone on the encountered edges is what contributed to its superior performance.

I. CONCLUSION

In this paper, we proposed an unconventional approach to initializing the pheromone level in ACO algorithms, which diverges from the standard pheromone initialization that has been applied in the literature. We categorize our algorithm as being a local approach while we refer to the standard algorithms as global approaches. Properly initializing the pheromone level on edges using a value close to the pheromone expected to be deposited by ants guarantees the system's convergence to the optimal solution in a reasonable amount of time. The goal of our local pheromone initialization approach is to make use of the information locally available to ants to calculate the initial pheromone level on uninitialized edges. We believe that our approach refines the initialization process since we do not expect all edges to have equal amount of pheromones to be deposited on them. Another advantage of locally initializing the pheromone level on edges is that it does not require the initialization of all the edges in the system, which we believe can be useful in situations where the solution space is huge and the optimal (or sub-optimal) solutions reside in a certain area of that space. Analyzing such information has the potential of providing further insight about the problem at hand.

We tested our initialization technique by incorporating it into the ACS algorithm to solve the TSP. We tested a variety of datasets of different sizes and compared the results to different versions of ACS employing standard pheromone initialization techniques. In general, our experiments show that local pheromone initialization can improve the results either by successfully constructing shorter tours or in cases in which it does not, it accomplishes a balance of achieving reasonable results in a fewer number of iterations. The initial experiments' results also show that increasing the number of ants in the system can further enhance the results. However, the number of cities (nodes) in the problem and the length of distances did not seem to have an impact on the quality of results obtained.

Based on the results of our experiments and the comparisons performed, we can conclude that our proposed local pheromone initialization technique generally provides better system performance when incorporated into ACO algorithms. Although the approach was tested on ACS to solve the TSP, but the results show great potential for the approach to be adopted by other ACO algorithms to solve other similar optimization problems.

REFERENCES

[1] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21-27, 1967.

[2] J. L. Deneubourg, "The self-organizing exploratory pattern of the Argentine ant," *Journal of Insect Behavior*, vol. 3, pp. 159-168, 1990.

[3] L. Deneubourg, S. Goss, N. Frank, A. Sendova-Franks, C. Detrain, and L. Cherétien, "The dynamics of collective sorting: Robot-like ants and ant-like robots," in *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From animals to animats*, Cambridge, MA, 1991, pp. 356-365.

[4] M. Dorigo, "Learning and natural algorithms," PhD. dissertation, Politecnico di Milano, Milan, Italy, 1992.

[5] M. Dorigo, "Ant algorithms solve difficult optimization problems," in *Proceedings of the 6th Euorpean Conference on Artificial Life*, Berlin, 2001, pp. 11-22.

[6] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Generation Computer Systems*, vol. 16, pp. 851-871, 2000.

[7] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, pp. 137-172, 1999.

[8] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the travelling salesman problem," *IEEE Transaction on Evolutionary Computation*, vol. 1, pp. 53-66, 1997.

[9] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: An autocatalytic optimizing process," Milan, Italy, 1991.

[10] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 26, pp. 29-41, 1996.

[11] M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, 2004.

[12] M. Dorigo and T. Stützle, "The ant colony optimization metaheuristic: Algorithms, applications and advances," in *Handbook of Metaheuristics*, Norwell, MA, 2002, pp. 251-285.

[13] M. Dorigo, T. Stützle, and G. Di Caro, "Special issue on Ant Algorithms," *Future Generation Computer Systems*, vol. 17, pp. 851-956, 2000.

[14] D. S. Johnson and L. A. McGeoch, "The travelling salesman problem: A case study in local optimization," in *Local Search in Combinatorial Optimization*, Chichester, UK, 1997, pp. 215-310.

[15] E. L. Lawler *et al.*, *The Travelling Salesman Problem*, Chichester, UK, 1985.

[16] G. Reinelt, *The travelling salesman: Computational solutions for TSP applications*. Berlin, Germany, Springer-Verlag, 1994.

[17] T. Stützle, "Local search algorithms for combinatorial problems: Analysis, improvements, and new applications," *Dissertations in Artificial Intelligence*, Sankt Augustin, Germany, 1999, vol. 220, pp. 203.

[18] T. Stützle and H. H. Hoos, "The MAX-MIN ant system and local search for the travelling salesman problem," in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, 1997, pp. 309-314.

[19] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generation Computer Systems*, 2000, vol. 16, pp. 889-914.