Regular Article

Solving the Traveling Salesman Problem with Ant Colony Optimization: A Revisit and New Efficient Algorithms

Hoang Xuan Huan[1], Nguyen Linh-Trung[1], Do Duc Dong[2], Huu-Tue Huynh[3]

[1] University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
[2] Institute of Information Technology, Vietnam National University, Hanoi, Vietnam
[3] International University, Vietnam National University, Ho Chi Minh City, Vietnam

Correspondence: Nguyen Linh-Trung, linhtrung@vnu.edu.vn

Abstract– Ant colony optimization (ACO) techniques are known to be efficient for combinatorial optimization. The traveling salesman problem (TSP) is the benchmark used for testing new combinatoric optimization algorithms. This paper revisits the application of ACO techniques to the TSP and discuss some general aspects of ACO that have been previously overlooked. In fact, it is observed that the solution length does not reflect exactly the quality of a particular edge belong to the solution, but it is only used for relatively evaluating whether the edge is good or bad in the process of reinforcement learning. Based on this observation, we propose two algorithms– Smoothed Max-Min Ant System and Three-Level Ant System– which not only can be easily implemented but also provide better performance, as compared to the well-known Max-Min Ant System. The performance is evaluated by numerical simulation using benchmark datasets.

Keywords– Ant colony optimization, traveling salesman problem, pheromone update rule, reinforcement learning, random walk.

1 Introduction

Ant Colony Optimization (ACO) is a randomized heuristic search method for solving NP-hard problems in combinatorial optimization. In principle, it imitates the biological behavior of a real ant colony when the ants try to find the shortest path between their nest and a food source. When foraging, ants deposit on the ground a substance called *pheromone*. At any time, an ant tends to follow a path on which are the pheromone trails left by previous ants. Ultimately, the ants will be able to find the shortest path from the food source to the nest, and the whole colony will follow this path to transport the food back to their nest.

Inspired by this biological behavior of real ant colony, ACO is then developed for "artificial ants". This is done by formulating the original optimization problem to the shortest-path problem over a graph associated with the former problem. ACO combines two types of information: *heuristic information*, which is related to the length of an edge of the graph, and *reinforcement learning information*, which is related to the local information represented by the intensity of pheromone trail on this edge. The pheromone intensity changes with time. For example, it could either be reduced (vaporized) if no other ants visit the edge or be enhanced (reinforced) if other ants leave more pheromone on the edge. As such, a rule or a set of rules is established for updating the pheromone intensity. The update rules in a specific ACO algorithm represent a search strategy of the algorithm and, hence, influence its efficiency.

ACO algorithms have been successfully applied, with high efficiency, to a wide range of different combinatorial optimization problems. The first ACO algorithm, called the *Ant System* (AS), was proposed by Dorigo in [1] and [2] for solving the well-known Traveling Salesman Problem (TSP). In the TSP, a salesman takes a tour, which is a closed path, through a set of cities under a condition that all cities must be visited and each city is visited once only. The objective is to find the shortest tour; hence, a tour is also called a solution. The TSP is so important so as to have become a benchmark for evaluating the effectiveness of a new algorithm. In this paper, we will use the TSP for comparing the algorithms. It is well known that ACO for the TSP is typically not well performing without a local search. However, in this work, we focus only on the behavior of ACO and, hence, isolate ACO from the local search in order to understand the behavior. If one is interested in solving the TSP using ACO and comparing it with other algorithms, the local search should be combined with ACO.

Many variants of AS have been developed (see for examples [1–9]), among which the Ant Colony System (ACS) [5] and the Max-Min Ant System (MMAS) [9] are the two most popular algorithms. There have also been various theoretical studies on the convergence and other characteristics of ACO algorithms (see [10–14]). These studies give guidance on how to select relevant parameters that are useful for improving the efficiency of an algorithm. It can be seen that the state-of-the-art ACO algorithms follow a main stream wherein

the length of a solution is considered important in performing updates of the pheromone intensity on an edge. In turns, the quality function of an algorithm, specifying how the pheromone intensity is adjusted in each update, is often formulated to be dependent on the solution length. For example, the quality function in the TSP problem is inversely proportional to the solution length. However, for new problems other than the TSP, experimental studies often need to be carried out carefully for determining a suitable model for the quality function as well as for predefining the upper and lower bounds constraining the pheromone intensity, in order to make the algorithm implementation efficient.

In this paper, by analyzing the variation tendency of the pheromone intensity and the quality of an edge with respect to the length of a solution to which this edge belongs, we observe that the solution length does *not* reflect exactly the quality of the local information of the edge. Hence, unlike the above-mentioned main stream in the ACO literature, we advocate for not using the solution length in updating the pheromone intensity on an edge at each iteration but only using it for relatively evaluating whether the edge is good or bad in the process of reinforcement learning. Experimentally, this will help avoid the burden involved in determining the quality function as well as the lower and upper bounds on the pheromone intensity. Based on this observation, we propose two new update rules, which are called the *Smoothed Max-Min Ant System (SMMAS)* and the *Three-Level Ant System (3-LAS)*, wherein the solution length is now not used for updating but only for relatively comparing the solutions constructed in order to find the best solution. In addition, the bounds in these algorithms are predefined in a relative manner rather an absolute one. 3-LAS is a further improvement on SMMAS by the introduction of a new parameter between the bounds in order to enhance edge classification. As a result, the implementation of these rules are much easier than those in AS, ACS or MMAS. In addition, experimental results show that SMMAS and 3-LAS outperform MMAS.

The paper is organized as follows. Section 2 introduces the general procedure for ACO used for solving the TSP and the update rules in AS, ACS and MMAS. Section 3 provides some mathematical analysis and discussions on the variation tendency of pheromone intensity and the effect of the solution length on the edge quality in local updates. Section 4 gives details on the proposed algorithms, SMMAS and 3-LAS. Section 5 shows numerical simulation results, comparing these algorithms with MMAS. Section 6 concludes the paper.

2 ACO Algorithms for Solving the TSP

2.1 The Traveling Salesman Problem

Consider a set of N cities whereby the lengths of connections between pairs of cities are known a priori. Starting from a particular city, a salesman traverses through all other cities and returns to the starting city under the condition that to each city he visits once

---

Procedure 1 General procedure for ACO algorithms

1) Initialize:
   number of ants (m), number of iterations (T), and other relevant parameters
   set pheromone intensity $\tau_{ij} = \tau_0$ for all $e_{ij} \in \mathcal{E}$
2) **for** $t = 1 : T$ **do**
3)   Construct m solutions (for m ants) using random walk
4)   Adjust the pheromone intensity using update rules
5) **end for**
6) Obtain the best solution $s^*(T)$

---

only. The objective of the TSP is to find the shortest tour taken by the salesman. The general TSP can be described in graph terms as follows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a complete graph, where $\mathcal{V}$ is the set of N vertices and $\mathcal{E}$ is the set of directed or undirected edges fully connecting the vertices. For simplicity, we only consider the so called simple graph, that is, each pair of vertices i and j is connected by only one edge. Denote by e_{ij} the edge connecting i and j. Each e_{ij} is associated with a positive weight, called the edge length and denoted by l_{ij}. If there exists at least one e_{ij} such that $l_{ij} \neq l_{ji}$, then the problem is said to be the asymmetric TSP (ATSP). A tour is often called a solution, and the tour with the shortest length is the optimal solution.

2.2 General Procedure for ACO Algorithms

Assume that there are m ants and each ant can determine how far it is away from a city. The ants also endow with a working memory for memorizing the cities they have visited. The practical procedure of an ACO algorithm, as summarized in Procedure 1, is of an iterative nature and performs a finite number of iterations, T. At initialization, set the values of the pheromone intensity on every edge in $\mathcal{E}$ to τ_0, with $\tau_0 > 0$, and pre-define T and other relevant parameters.

Next, for each iteration t, with $0 < t \leq T$, all m ants are first randomly placed in the cities. Then, each ant k, with $1 \leq k \leq m$, constructs its own solution, $s_k(t)$, step-by-step by repeating the procedure of random walk, as presented in Section 2.3. The lengths of all the constructed solutions are then compared and, next, used for updating the pheromone intensity on edges.

Let $s^*(t)$ be the best solution constructed up to iteration t. Then, after T iterations, we obtain $s^*(T)$ and call it the "good-enough" solution of the TSP. Note that, because T is finite, the "good-enough" solution might not be the optimal solution but be good enough for practical purposes. The number of solutions, S, that we want to construct is $S = mT$. Often, one uses the same S to compare different algorithms although the values of m used in the algorithms may be different.

2.3 Procedure of Random Walk

Within an iteration and at each step in the process of constructing a solution, an ant located at some city

makes a probabilistic choice for going to another city to which this ant has not previously visited. This choice depends on the pheromone intensity on the edge and the heuristic information of the edge, which are locally available. Generally, an ant prefers paths which have strong pheromone intensity, and cities which are close to it. Mathematically put, at iteration t, an ant k located in city i chooses to go to city j with a transitional probability of

$$P_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta}{\sum\limits_{u \in J_{ki}} \tau_{iu}^\alpha(t)\eta_{iu}^\beta}, & j \in J_{ki}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $\tau_{ij}(t)$ is the pheromone intensity on edge e_{ij}, η_{ij} is the heuristic information ($\eta_{ij} = 1/l_{ij}$), J_{ki} is the set of cities to which this ant has not previously visited, α and β are parameters that determine the relative importance of the pheromone intensity and the heuristic information. Note that when applied to ACS the transitional probability defined in (1) has some slight modifications. These modifications, however, do not affect the analysis and discussion provided in Section 3. Hence, we skip them for the simplicity of presentation.

2.4 Important Pheromone Update Rules

2.4.1 Ant System (AS): At iteration t, when step (3) in Procedure 1 has completed, the pheromone intensity on each edge e_{ij} is updated by the following update rule:

$$\tau_{ij} \leftarrow \rho\tau_{ij} + \Delta\tau_{ij}, \tag{2}$$

with the pheromone deposit, $\Delta\tau_{ij}$, for AS being defined as

$$\Delta\tau_{ij}^{(AS)} = \sum_{k=1}^{m} \nu_k, \tag{3}$$

where $\nu_k = 1/L_k$ if $e_{ij} \in s_k(t)$ and $\nu_k = 0$ otherwise; L_k is the length of solution $s_k(t)$. The update rule described by (2) is general for all ACO algorithms while the rule by (3) is specific for AS. The general update rule in (2) includes two types of adjustment to the pheromone intensity on e_{ij}. The first adjustment is due to the natural evaporation of the pheromone intensity updated in the previous iteration; the intensity is reduced by the evaporation factor, ρ, with $0 < \rho < 1$. The second is the pheromone deposit, $\Delta\tau_{ij}^{(AS)}$, on e_{ij} which is newly added when this edge was traversed by some ants.

2.4.2 Ant Colony System (ACS): In ACS, the pheromone update rules expressed by (2) and (3) are modified to change both locally and globally.

2.4.2.1 Local update: The local update rule is applied to edges which have been visited by the ants during iteration t. In the process of constructing its own solution, when an ant traverses through e_{ij} it changes the pheromone intensity on this edge locally by applying the following update rule:

$$\tau_{ij} \leftarrow \rho\tau_{ij} + (1-\rho)\tau_1, \tag{4}$$

where τ_1 is a predefined positive constant. At initialization, the pheromone intensity is set to be $\tau_0 = 1/(nL_{nn})$, where L_{nn} is the solution length found by the nearest neighbor heuristic method in [15]. Note that, in the original ACS [6], τ_0 and ξ are used in (4) in replacement of τ_1 and ρ, respectively. However, since they do not affect the analysis and discussion in Section 3, we just use τ_1 and ρ here for the simplicity of presentation.

2.4.2.2 Global update: The global update rule is applied only to the edges belonging to the best solution, $s^*(t)$, at the corresponding iteration, t. This best solution is either called "globally-best" (gb) if it is the best for all iterations up to t, or "iteration-best" (ib) if it is the best for iteration t. Once all solutions have been constructed for iteration t, the pheromone intensity on all $e_{ij} \in s^*(t)$ are updated according to the general update rule (2), with the pheromone deposit, $\Delta\tau_{ij}$, for ACS being defined as

$$\Delta\tau_{ij}^{(ACS)} = (1-\rho)\frac{1}{L_{gb}}, \tag{5}$$

where L_{gb} is the length of $s^*(t)$.

2.4.3 Max-Min Ant System (MMAS): MMAS is a direct improvement on AS with the following modifications. Firstly, like the global update rule in ACS, once all solutions have been constructed at an iteration, only edges which belong to the best solution are updated using the general update rule in (2), where the pheromone deposit, $\Delta\tau_{ij}$, for MMAS is given by

$$\Delta\tau_{ij}^{(MMAS)} = \begin{cases} (1-\rho)\frac{1}{L_{gb}}, & \text{if } e_{ij} \in s^*(t), \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

Secondly, the pheromone intensity τ_{ij} on e_{ij} is constrained in the interval $[\tau_{min}, \tau_{max}]$, where τ_{min} and τ_{max} are predefined non-zero parameters. These non-zero parameters are used to reduce *stagnation*– a situation in which all the ants follow the same path and construct the same solution but not yet optimal. Once the above global update has been performed and if τ_{ij} falls outside of $[\tau_{min}, \tau_{max}]$, then τ_{ij} is re-adjusted by the following:

$$\tau_{ij} = \begin{cases} \tau_{min}, & \text{if } \tau_{ij} < \tau_{min}, \\ \tau_{max}, & \text{if } \tau_{ij} > \tau_{max}. \end{cases} \tag{7}$$

The choices of τ_{min} and τ_{max} greatly influence the efficiency of the algorithm. Thirdly, *re-initialization* is used in MMAS when stagnation occurs. Fourthly, when the algorithm is near convergence, *smoothing* the change of pheromone intensity is then used in order to enhance the *exploration*, which extends the search to new solutions for finding more good solutions. This smoothing replaces the rule defined in (6) with the following update rule:

$$\tau_{ij} \leftarrow \tau_{ij} + \delta(\tau_{max} - \tau_{ij}), \tag{8}$$

where δ is a constant, with $0 < \delta < 1$.

Despite the fact that ACS and MMAS have been well analyzed (e.g., in [9]), we will focus on the behavior of the pheromone intensity in these algorithms in the next section. The analysis will serve as a basis for our later proposing new algorithms.

3 Pheromone Variation Tendency and Effect of Solution Length on Edge Quality

In the TSP, like in other NP-hard combinatorial optimization problems, we can perform an exhaustive search to obtain the optimal solution in principle . However, exhaustive searching is not desirable in practice because when the search space is too large we could possibly not be able to search the whole space due to limited resources. To reduce the search space, ACO makes use of heuristic information, related to the length of an edge, and applies reinforcement learning, locally related to the pheromone intensity on the edge. In other words, at each iteration, instead of performing a random search over the entire set of admissible solutions, ants prefer to search for edges which are short in length and strong in pheromone intensity, as modeled by the transitional probability defined in (1); hence, the search space is reduced.

A randomized search based on the heuristic information makes the search be more dynamic and flexible on a space larger than that in other methods based on heuristic information. As a result, a better solution or even the optimal solution may be obtained. In addition, when further combined with reinforcement learning, the size of the search space is gradually reduced without eliminating the majority of good solutions; hence, the algorithm performance is improved. Therefore, the performance of an ACO algorithm largely depends on the rules defined for pheromone updating and their associated parameters.

Various theoretical and experimental studies have been carried out in order to better understand ACO algorithms, to improve their efficiency as well as to examine ways of selecting algorithm parameters. However, several questions are of concern, after the update rules have been applied. First, when the search space is reduced, the search action might skip good solutions which do not belong to the reduced search space; that means the degree of exploration is reduced. Whereas, if we want to increase the exploration then we need to enlarge the search space. How do we define a good update rule in order to compromise between the exploration and the size of the search space? Second, an edge belongs to many solutions having different lengths, and the pheromone intensity update for this edge at each iteration randomly depends on the solutions that the ants constructed. How does the relationship between local information and global information affect the algorithm efficiency? Third, in each update, how much of adjustment in the pheromone intensity update is appropriate? In the following, we will provide more insights toward these questions.

3.1 Variation Tendency of Pheromone Trails

In the TSP, there always exists an optimal solution because the number of admissible solutions is finite. Let L_{opt} be the length of the optimal solution. It can be easily verified that for ACS the pheromone intensity on e_{ij} always satisfies the following inequality

$$\tau_{ij} \geq \min\{\tau_0, \tau_1, \frac{1}{L_{opt}}\}. \tag{9}$$

The right-hand side of (9) is set to be τ_{min}. Denote by $p_s(k,t)$ the probability that ant k constructs solution $s_k(t)$ at iteration t. For both ACS and MMAS, it has been shown that there exists a positive constant p_0 such that $p_s(k,t) > p_0$ for all $s_k(t)$, k, and t [11]. This means that *all* edges have an opportunity to be chosen by the ants.

In terms of pheromone convergence, some fundamental results were established by Gutjahr in [10], and by Stützle and Dorigo in [11]. Specifically, by using Markov models, Gutjahr provided a proof for convergence in probability for a rather large class of ACO algorithms, including MMAS. Stützle and Dorigo considered the convergence for ACS provided that the optimal solution has been found. Even though these fundamental results are significant, one can only implement a *finite* number of iterations in practice. Hence, it is still desirable to have more insights about the behavior of pheromone intensity in order to improve the algorithm performance.

Starting from a big enough iteration, the pheromone intensity on most edges quickly becomes small and, hence, the algorithm focuses on searching around the best constructed solution [16]. This implies that ants tend to update around the edges that have strong pheromone intensity (near τ_{max} in the case of MMAS). The pheromone update rules and other results in [10] and [11] give the variation tendency of pheromone intensity on the edges that belong to the best solution. The following proposition provides the tendency on the edges that do *not* belong to the best solution, for ACS and MMAS.

Proposition 1. *Assume that edge e_{ij} belongs to an admissible solution $s(t)$ and that there exists t_0 such that e_{ij} does not belong to the best solution $s^*(t)$ for all $t \geq t_0$. The following assertions are true:*

1) *If the ACS update rule is applied then $\tau_{ij}(t)$ converges in probability to τ_1;*
2) *If the MMAS update rule is applied then $\tau_{ij}(t) = \tau_{min}$ for all t satisfying*

$$t > t_0 + \frac{\ln(\tau_{min}/L_{opt})}{\ln(\rho)}. \tag{10}$$

Proposition 1 shows that the pheromone intensity on the edges which do not belong to the best solution converges to τ_1 in the case of ACS, and to τ_{min} in the case of MMAS. For MMAS, this proposition indicates that the pheromone intensity on edges which do not belong to the best solution *quickly* reduces to τ_{min}; in other words, it implies that the exploration decreases quickly. In addition, this proposition re-confirms the variation tendency of pheromone intensity for ACS, which was stated by Stützle and Dorigo in [11] under the assumption that the optimal solution has been found. In the following, we will explain in more detail
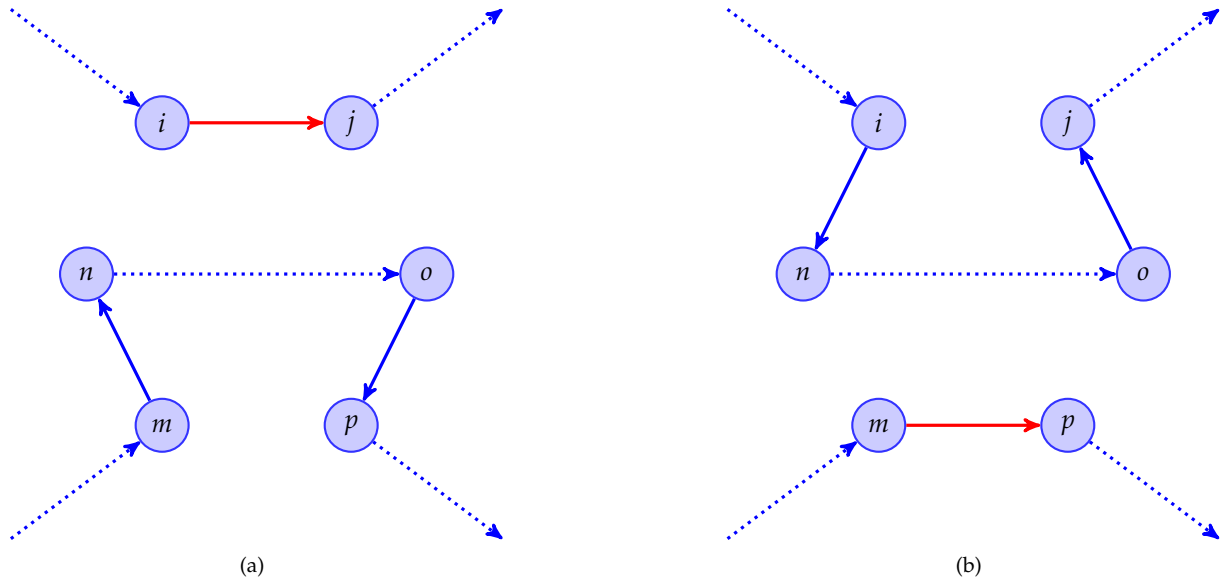
Figure 1.   (a)– solution through e_{ij} but e_{mp}; (b) – solution through e_{mp} but e_{ij}. The two solutions differ by three edges.

how ACS and MMAS deal with exploration enhancement. Later, in Section 4, we will propose some new updating rules to compromise between the exploration and the intensification in reinforcement learning. Note that intensification is a search strategy that focuses the search to perform around the best solution.

In ACS, the use of the global update in reinforcement causes searching around the best solution. Equation (9) shows that it reduces stagnation. Hence, the efficiency of ACS is improved, as compared to AS. However, it also reduces the exploration when using the global-best updating, and experimental studies suggest that iteration-best updating is better. In addition, unlike AS, the local update rule in ACS reduces the pheromone intensity along the solution that was not good while its pheromone intensity had been previously increased. In an indirect way, it increases the exploration at edges which are not often used by the ant. However, this expansion causes the search space to become too large and, thus, limits the algorithm efficiency.

In MMAS, the constraint of pheromone intensity to be in $[\tau_{\min}, \tau_{\max}]$ helps avoid stagnation and enlarge the diversification (i.e., exploration of new search regions through the introduction of new attribute combinations), in comparison with AS and ACS. Setting τ_0 to be $\tau_{\max}$ and m to be large makes the pheromone intensity on *bad* edges slowly decreased and the diversification increased. In addition, the re-initialization of the pheromone intensity to be $\tau_{\max}$, whenever the pheromone intensity on most of the edges which do not belong to $s^*(t)$ is very close to $\tau_{\min}$, also enhances the diversification. Therefore, MMAS is more efficient than ACS. However, a convenient way to pre-define $\tau_{\min}$ and $\tau_{\max}$ depends on each specific optimization problem and finding it is still an open issue.

3.2 Effect of Solution Length on Edge Quality

While constructing a solution, the probabilistic choice by which an ant decides to move from one city to the

next is biased by the local information of this edge in terms of its pheromone intensity and length. However, only the edges which belong to the best solution have their pheromone intensity increased. Naturally, an edge is considered bad if the length of the solution to which it belongs is long, and considered good, otherwise.

In an undirected complete graph with N vertices, there are $(N-2)!$ solutions that consists of a particular edge. This gives rise to the following question: Does the length of a solution to which an edge belongs reflect exactly the quality of the local information of this edge? The following proposition will provide an answer to this question.

Proposition 2. *Let e_{ij} and e_{mp} be two different edges of an undirected complete graph. For any arbitrary Hamilton circle which goes through e_{ij} but e_{mp}, there always exists another Hamilton circle that goes through e_{mp} but e_{ij} such that these two circles have at most three different edges.*

This proposition can be illustrated in Figure 1. Figure 1(a) presents the solution that includes e_{ij} but e_{mp}. Since, m cannot connect to p, there must exist two vertices n and o forming the edges e_{mn} and e_{op}. In addition, there may be a number of edges between n and o. A solid arrow indicates an edge between two vertices and a dotted arrow indicates that there may be a number of concatenated edges in between. Now, we implement a change of the solution in Figure 1(a) to the solution in Figure 1(b) such that the latter solution includes e_{mp} but e_{ij}. Since m is now directly connected with p, there exists no longer edges e_{mn} and e_{op}. Also, since there is no longer edge e_{ij}, i must be connected with another vertex, and j similarly. For a minimum number of changes between the two solutions, one can let i be connected with n and o with j, thus, forming edges e_{in} and e_{oj}. Clearly, the two solutions differ by three edges: (e_{ij}, e_{mn}, e_{op}) in the first solution while (e_{in}, e_{oj}, e_{mp}) in the second one. This proposition suggests us to revisit the issue of how to evaluate the quality of an

edge in ACO algorithms, as in the following discussion.

3.2.1 On relationship between local and global information: While constructing a solution by the procedure of random walk, each ant uses the local information of the unvisited edges but the decision for updating the pheromone intensity depends on the relative comparison among the lengths of the constructed solutions. Proposition 2 indicates that, when there is a large number of vertices, the length of each edge is very small in comparison to the length of a solution to which this edge belongs, hence, it takes a very weak role in the quality of the solution. Therefore, at each iteration, an edge may be a component of a good solution or bad one by chance and, therefore, it is evaluated to be good or bad also by chance. After a number of iterations, the pheromone intensity on a bad edge quickly becomes very small and, hence, the edge will almost be eliminated out of search space. This is a limitation of ACO algorithms.

3.2.2 Quality of updated pheromone deposits: In ACS and MMAS, the updated pheromone deposit on each edge depends on the length of the solution to which this edge belongs. Proposition 2 indicates that this dependency is complicated and unnecessary. Indeed, the length of the solution is mainly used for evaluating if the edge is good or bad in a relative manner.

4 PROPOSED ALGORITHMS

Based on the analysis in Section 3, we propose two ACO algorithms in this section. Specifically, these algorithms differ from state-of-the-art algorithms in two main points. First, the solution length is not used for updating the pheromone deposit. Second, the lower and upper bounds, $\tau_{\max}$ and $\tau_{\min}$, are not defined in an absolute manner; instead, they are defined in the form of the ratio $\tau_{\max}/\tau_{\min}$. In our experience, this ratio is set by

$$\frac{\tau_{\max}}{\tau_{\min}} = Nk, \quad \frac{\tau_{\mathrm{mid}}}{\tau_{\min}} = k, \tag{11}$$

where

$$k = \begin{cases} (N+50)/100, & \text{for } N \geq 50, \\ 1, & \text{otherwise.} \end{cases} \tag{12}$$

The choices in (11) are based on ad-hoc consideration of the algorithm complexity. In fact, we wish to have $\tau_{\max}/\tau_{\min}$ proportional to the possible number of tours while $\tau_{\mathrm{mid}}/\tau_{\min}$ is this proportional constant k. Following the architecture of our algorithms, our choice seems minimizing their complexity. This point will be illustrated by the simulated results in Section 5. Furthermore, the number of vertices in each instance is greater than or equal to 50; hence, our best linear estimate for k is as given in (12).

4.1 SMMAS (Smooth Max-Min Ant System)

As by (8), Stützle and Hoos [9] have suggested a way to smooth the pheromone intensity in order to increase the exploration of MMAS when the algorithm is near

convergence. The solution construction in SMMAS is the same as in MMAS. The improvement, as compared to MMAS, is done by using the general update rule defined by (2) with the following modification to the update rule for the pheromone deposit

$$\Delta\tau_{ij}^{(\mathrm{SMMAS})} = \begin{cases} (1-\rho)\tau_{\max}, & \text{if } e_{ij} \in s^*(t), \\ (1-\rho)\tau_{\min}, & \text{otherwise.} \end{cases} \tag{13}$$

Together with pre-defining the ratio of $\tau_{\max}/\tau_{\min}$, this update rule simplifies the implementation of the algorithm and avoids difficulty in determining the absolute values for $\tau_{\min}$ and $\tau_{\max}$. In comparison with MMAS, the pheromone intensity updated in SMMAS changes more slowly and always remains in the interval $[\tau_{\min}, \tau_{\max}]$, hence less computation is executed in the algorithm.

4.2 Three-level Ant System (3-LAS)

In [7], an algorithm called Multiple-Level Ant System (MLAS) was proposed to improve on both ACS and MMAS. Inspired by MLAS, 3-LAS also employs a new parameter in the interval $[\tau_{\min}, \tau_{\max}]$, denoted by τ_{mid}, in order to enhance edge classification. The solution construction in 3-LAS is the same as in MMAS. The update rule is improved, combining the advantages of SMMAS and MLAS, by using the general rule (2) with the pheromone deposit being modified according to the following rule:

$$\Delta\tau_{ij}^{(\text{3-LAS})} = \begin{cases} (1-\rho)\tau_{\max}, & \text{if } e_{ij} \in s^*(t), \\ (1-\rho)\tau_{\mathrm{mid}}, & \text{if } e_{ij} \notin s^*(t) \text{ and} \\ & \quad \exists \text{ an ant uses it,} \\ (1-\rho)\tau_{\min}, & \text{otherwise.} \end{cases} \tag{14}$$

Note that when $\tau_{\mathrm{mid}} = \tau_{\min}$, 3-LAS becomes SMMAS. In addition to using $\tau_{\max}/\tau_{\min}$, we also pre-define the ratio $\tau_{\mathrm{mid}}/\tau_{\min}$ rather than the absolute value for τ_{mid}. Hence, similar to SMMAS, this simplifies the implementation of the algorithm. This update rule is simpler than ACS and has the same algorithmic complexity.

5 SIMULATION

To evaluate the performance of the algorithm, we numerically compare it with MMAS (without using local search) using 12 benchmark datasets extracted from `TSPLIB95` [17]: 8 datasets for the TSP (`ei51`, `kroA100`, `kroB150`, `d198`, `kroA200`, `lin318`, `att532` and `rat783`) and 4 datasets for the ATSP. These datasets have been tested in [9] and with two additional datasets `kroB150`, `kroA200` to increase the confidence in parameter selection. Experimental studies for MMAS were run by the software package `ACOTSP` (version 1.0), developed by Stützle [18], for the first 8 datasets. The results are better than those published in [9]. `ACOTSP` cannot be used for the ATSP datasets [9] so we do not have data for the second row in each of these problems. Note that, there have been many simulations done for large numbers of instances and large ranges for parameters. Given the fact that we are interested in

Table I
SOLUTIONS FOUND BY MMAS, SMMAS AND 3-LAS

dataset	MMAS		SMMAS		3-LAS	
eil51	426.44 (0.10%)		**426 (0%)**		426.2 (0.05%)	
opt: 426	**426**	428	**426**	426	**426**	427
kroA100	21304.4 (0.11%)		21293.44 (0.05%)		**21283.12 (0.01%)**	
opt: 21282	**21282**	21378	**21282**	21379	**21282**	21296
kroB150	26315.72 (0.71%)		26142.28 (0.05%)		**26136.44 (0.02%)**	
opt: 26130	26176	26438	**26130**	26181	**26130**	26147
d198	15950.96 (1.08%)		15954.04 (1.1%)		**15944.52 (1.04%)**	
opt: 15780	15875	16034	15884	16005	**15841**	16014
kroA200	29665.84 (1.01%)		29436.56 (0.23%)		**29431.88 (0.22%)**	
opt: 29368	29422	29843	**29394**	29540	**29394**	29546
lin318	42956.96 (2.21%)		42260.48 (0.55%)		**42237.68 (0.5%)**	
opt: 42029	42837	43324	42136	42373	**42091**	42512
att532	28767.1 (3.9%)		28113.08 (1.54%)		**28096.44 (1.48%)**	
opt: 27686	28636	28920	**27885**	28683	27888	28240
rat783	9283.6 (5.42%)		**8949 (1.62%)**		9260.12 (5.16%)	
opt: 8806	9249	9336	**8901**	8998	8946	9539
ry48p	14523.4 (0.7%)		14459.08 (0.26%)		**14429.44 (0.05%)**	
opt: 14422	N/A		**14422**	14532	**14422**	14460
ft70	38922.7 (0.65%)		38920.48 (0.64%)		**38825.2 (0.39%)**	
opt: 38673	N/A		**38707**	39268	**38707**	39238
kro124p	36573.6 (0.95%)		36566.8 (0.93%)		**36445.08 (0.59%)**	
opt: 36230	N/A		**36230**	36829	**36230**	36586
ftv170	2817.7 (2.28%)		2810.8 (2.03%)		**2804.04 (1.78%)**	
opt: 2755	N/A		**2790**	2854	**2790**	2838

improving MMAS and ACS as well as understanding the behavior of ACO instead of comparing different algorithms for solving the TSP, in our opinion it suffices to take as benchmark experiment realized by [9].

Each algorithm was run 25 times with the number of solutions of $S = 10000 \times N$ for the TSP datasets and of $S = 20000 \times N$ for the ATSP datasets, as done in [9]. Other parameters: $\alpha = 1$ and $\beta = 2$ as in [9], $m = N/2$ and $\rho = 0.02$. For 3-LAS, we additionally use $\tau_{mid}/\tau_{min} = k$. In the simulation, re-initialization as in MMAS was used to increase the exploration.

The experiment results are presented in Table I. For a particular algorithm and dataset, the top figure represents the average length of the best found solutions. Next to it is the percentage figure in bracket indicating the deviation from the optimal value (denoted as "opt" in Table I). The bottom left and right figures represent the best and the worst solutions in all run-times. Bold figures indicate the best results across the implemented algorithms for a particular dataset. Results for the 4 ATSP datasets are shown in the bottom rows.

It can be seen that the average values correctly reflect the efficiency of the algorithm while the best and worst values are used as reference for the dynamic range of the solutions found by the algorithm. In regard to the average values, it can be seen that, over the 12 tests, 3-LAS gives better results than SMMAS does, except for tests eil51 and rat783, and SMMAS is better than MMAS. Considering the best values, the results show that all the algorithms yield the optimal values for tests eil51 and kroA100. For the remaining tests (in the 8 TSP tests), SMMAS and 3-LAS give better results than MMAS does. Also, for test kroB150,

both SMMAS and 3-LAS found the optimal results but MMAS. Therefore, we can conclude that, while both proposed algorithms are simple and easy to use, they also provide better efficiency than MMAS, and 3-LAS seems to be dominating.

6 CONCLUSIONS

ACO algorithms are efficiently used in combinatorial optimization, in which the TSP is an important problem. There have been various theoretical studies on the characteristics of the algorithms and various pheromone update rules have been developed. For the two most popular ACO algorithms – ACS and MMAS–, we have analyzed and discussed the variation tendency of the pheromone intensity and the edge quality with respect to the length of the solution constructed by an ant traversing this edge.

The key contribution of this paper is that, unlike the main stream of thought in the ACO literature, we have showed that the length of a solution does not reflect exactly the quality of a particular edge belonging to the solution, but it is only used for relatively evaluating whether the edge is good or bad in the process of reinforcement learning. Hence, the determination of the upper and lower bounds on the pheromone intensity does not need to depend on the quality function. Instead, it should only be dependent on the search strategy– intensification or diversification– which is then reflected by the ratio of the upper bound over the lower bound. This strategy simplifies the implementation of the algorithm, because we do not need to know the absolute values of these bounds.

Based on the above observation, we have proposed two new algorithms SMMAS and 3-LAS. These algorithms not only can be easily implemented but also provide better performance, as compared to MMAS. The performance was evaluated by numerical simulation using 12 benchmark datasets.

The scaling problem has been investigated thoroughly in [19, 20]. Blum and Dorigo [19] discussed a hyper cube frame work to limit the pheromone values between 0 and 1. This normalization is trivial while we think that it can be any interval $[\tau_{\min}, \tau_{\max}]$ under the condition that the ratio $\tau_{\max}/\tau_{\min}$ remains the same. Birattari et al. [20] showed that the ACO algorithm scaling, in particular ACS and MMAS, is invariant, under the condition that the performance index is proportional to any monotonically increasing function; this is because the pheromone trail updating does not depend on its value. Therefore, the invariance is not of any influence to be considered. Recently, the invariance discussed in the work of Zhang and Feng [21] is also applied to SMMAS and 3-LAS. However, their algorithms do not have the advantages of ours that have been discussed previously. All these points consolidate our point of view as well as reconfirm the superiority of our proposed algorithms.

Finally, our proposed algorithms are specially useful for new problems, other than the TSP, when experimental studies often need to be carried out carefully in determining a suitable model for the quality function and predefining the upper and lower bounds on the pheromone intensity. In contributing to this perspective, we have successfully applied the proposed algorithms to the haplotype inference problem in bioinformatics [22], as an example.

Appendix
Proof of Proposition 1

Assertion 1: For all $\epsilon > 0$, we need to prove that

$$\lim_{t \to \infty} P(|\tau_{ij}(t) - \tau_1| > \epsilon) = 0.$$

The local update rule in (4) can be rewritten as follows:

$$\tau_{ij} \leftarrow \rho \tau_{ij} + (1 - \rho)\tau_1 = \tau_1 + \rho(\tau_{ij} - \tau_1).$$

If, from the iteration t_0 to iteration t, the pheromone intensity on edge e_{ij} is locally updated k times then

$$\tau_{ij}(t) = \tau_1 + \rho^k [\tau_{ij}(t_0) - \tau_1].$$

Hence, there exists a large enough k_0 such that for all $k > k_0$ we have:

$$|\tau_{ij}(t) - \tau_1| = |\rho^k [\tau_{ij}(t_0) - \tau_1]| \leq \epsilon.$$

Since

$$\lim_{t \to \infty} P(|\tau_{ij}(t) - \tau_1| > \epsilon) = \lim_{t \to \infty} P(k \leq k_0),$$

we need to prove that

$$\lim_{t \to \infty} P(k \leq k_0) = 0.$$

Indeed, let $P_{ij}(r)$ be the probability that the pheromone intensity on edge e_{ij} is locally updated by ant r at each iteration. Similar to [11], we then have the following statement:

$$1 > P_{ij}(r) \geq p_0 > 0.$$

In other words, if we let $P_{\neg ij}(r)$ be the probability that the pheromone intensity on e_{ij} is *not* locally updated by ant r at each iteration, then $P_{\neg ij}(r)$ satisfies the following:

$$0 < P_{\neg ij}(r) \leq 1 - p_0 < 1. \tag{A.1}$$

Assume that $t = t_0 + q$. Then for iterations from t_0 to t, it can be seen that edge e_{ij} is locally updated by mq ants in a random manner where the probability of not updating satisfies (A.1) under any condition. By doing an analysis similar to the Bernoulli trial, we have

$$P(k \leq k_0) = \sum_{i=1}^{k_0} C_{mq}^i (1 - p_0)^{mq-i} \tag{A.2}$$

Therefore,

$$\lim_{t \to \infty} P(k \leq k_0) \leq \lim_{q \to \infty} \sum_{i=1}^{k_0} C_{mq}^i (1 - p_0)^{mq-i} = 0.$$

We remark that one can use Poisson distribution to estimate the probability of edge updated i times to have an estimate better than (A.2).

Assertion 2: With $t = t_0 + q$, we have:

$$\begin{aligned}
\tau_{\min} &\leq \tau_{ij}(t) \\
&= \max\{\rho^q \tau_{ij}(t_0), \tau_{\min}\} \\
&\leq \max\{\rho^q L_{\mathrm{opt}}^{-1}, \tau_{\min}\}
\end{aligned}$$

for all q such that

$$q \geq \frac{\ln(\tau_{\min}/L_{\mathrm{opt}})}{\ln(\rho)}.$$

References

[1] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: An autocatalytic optimizing process," Politecnico di Milano, Milano, Italy, Tech. Rep. 91-016, 1991.

[2] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Politecnico di Milano, Italy, 1992.

[3] R. Bullnheimer, B. amd Hartl and C. Strauss, "A new rank based version of the ant system - A computational study," *Central European Journal of Operations Research*, vol. 7, no. 1, pp. 25–38, 1999.

[4] M. Dorigo and T. Stützle, *Ant colony optimization*. MIT Press, 2004.

[5] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," vol. 1, no. 4, pp. 28–39, 2006.

[6] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[7] D. Q. Huy, D. D. Dong, and H. X. Huan, "Multi-level ant system - A new approach through the new pheromone update for ant colony optimization," in *Proc. 2006 Int. Conf. Research, Innovation and Vision for the Future (RIVF)*, 12-16 Feb. 2006, pp. 55–58.

[8] D. Do Duc, H. Dinh, and H. Hoang Xuan, "On the pheromone update rules of ant colony optimization approaches for the job shop scheduling problem," in *Proc.*

Pacific Rim Int. Workshop on Multi-Agents (PRIMA 2008), 2008, pp. 153–160.

[9] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generation Computer Systems*, vol. 16, no. 9, pp. 889–914, June 2000.

[10] W. J. Gutjahr, "ACO algorithms with guaranteed convergence to the optimal solution," *Information Processing Letters*, vol. 82, no. 3, pp. 145 – 153, 2002.

[11] T. Stützle and M. Dorigo, "A short convergence proof for a class of ant colony optimization algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 358–365, 2002.

[12] B. Doerr, F. Neumann, D. Sudholt, and C. Witt, "On the influence of pheromone updates in ACO algorithms," University of Dortmund, Germany, Tech. Rep. No. CI-223/07, 2006.

[13] Y. Zhou, "Runtime analysis of an ant colony optimization algorithm for TSP instances," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1083–1092, 2009.

[14] T. Kötzing, F. Neumann, H. Röglin, and C. Witt, "Theoretical properties of two ACO approaches for the traveling salesman problem," in *Proc. 7th Int. Conf. Swarm intelligence (ANTS 2010)*, Brussels, Belgium, Sep. 8-10 2010, pp. 324–335.

[15] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, "An analysis of several heuristics for the traveling salesman problem," *SIAM Journal on Computing*, vol. 6, no. 3, pp. 563–581, 1977.

[16] P. Pellegrini and A. Ellero, "The small world of pheromone trails," in *Proc. 6th Int. Conf. Ant Colony Optimization and Swarm Intelligence*, 2008, pp. 387–394.

[17] G. Reinelt. Tsplib. Heidelberg University. Germany. [Online]. Available: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

[18] T. Stützle. Acotsp. (Software package). [Online]. Available: http://www.aco-metaheuristic.org/aco-code/public-software.html

[19] C. Blum and M. Dorigo, "The hyper-cube framework for ant colony optimization," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 34, no. 2, pp. 1161–1172, 2004.

[20] M. Birattari, P. Pellegrini, and M. Dorigo, "On the invariance of ant colony optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 732–742, 2007.

[21] Z. Zhang and Z. Feng, "Two-stage updating pheromone for invariant ant colony optimization algorithm," *Expert Systems with Applications*, vol. 39, no. 1, pp. 706–712, 2012.

[22] D. D. Dong and H. X. Xuan, "A fast and efficient ant colony optimization for haplotype inference by pure parsimony," in *Proc. 3rd Int. Conf. Knowledge and Systems Engineering*, Hanoi, Vietnam, 14-17 Oct. 2011.

Nguyen Linh-Trung received both the B.Eng. and Ph.D. degrees in Electrical Engineering from Queensland University of Technology, Brisbane, Australia. From 2003 to 2005, he had been a postdoctoral research fellow at the French National Space Agency (CNES). In 2006, he joined the University of Engineering and Technology within Vietnam National University, Hanoi, and is currently an associate professor at its Faculty of Electronics and Telecommunications. He has held visiting positions at Telecom ParisTech, Vanderbilt University, Ecole Supérieure d'Electricité (Supelec) and the Université Paris 13 Sorbonne Paris Cité. His research focuses on methods and algorithms for data dimensionality reduction, with applications to biomedical engineering and wireless communications. The methods of interest include time-frequency analysis, blind source separation, compressed sensing, and network coding. He has published more than 40 scientific papers and a textbook on digital signal processing. He was co-chair of the technical program committee of the annual International Conference on Advanced Technologies for Communications (ATC) in 2011 and 2012. He is a senior member of the IEEE.

Do Duc Dong received the B.S., M.S. and Ph.D. degrees in Computer Sciences from the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, in 2004, 2008 and 2012, respectively. His interest is applying meta-heuristic approach to solve the optimization problems in large-scale data and high-performance computing context. Recently, he is interested in computational methods in data mining for the emerging biological data.

Huu-Tue Huynh was born in Hue, Viet Nam. He received the Sc.D. degree in 1972 from Laval University, Canada, where from 1969 to 2004 he was a faculty member of the Department of Electrical and Computer Engineering. In 2004, he left Laval University to become Chairman of the Department of Data Processing at The College of Technology of the Vietnam National University, Hanoi. During the period 2007–2011, he was the President of Bac Ha International University, Vietnam. He is now a research professor at the School of Electrical Engineering, International University, Vietnam National University, Ho Chi Minh City, Vietnam. He was an Invited Guest at The AT&T Information Systems in Neptune, N.J. in 1984 and has been invited to give lectures at several Universities in Europe, America as well as in Asia. Professor Huynh is author and coauthor of two books and more than two hundred papers in Information Processing. He has served as Consultant to a number of Canadian Government Agencies and Industries. His research interests cover stochastic simulation techniques, information processing, fast algorithms and architectures with applications to finance and to communications.

Hoang Xuan Huan was lecturer at the Faculty of Mathematics of Hanoi University from 1980 to 1995, where he obtained his Ph.D. degree with the work on "Separable calibration and globally minimal currents". In 1995, he has joined the Faculty of Information Technology of the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, where he is currently an associate professor. He has published 3 books and more than 50 papers, covering a large spectrum of topics including optimization techniques, evolutionary computations, economic mathematics, machine learning and bioinformatics.