Hybrid PACO with Enhanced Pheromone Initialization for Solving the Vehicle Routing Problem with Time Windows

Wei Shi*, Thomas Weise†‡, Raymond Chiong‡, and Bülent Çatay§

*Joint USTC-Birmingham Research Institute in Intelligent Computation and Its Applications (UBRI), School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China, 230027. Emails: shw1990@mail.ustc.edu.com, tweise@ustc.edu.cn

†Corresponding Author

‡Faculty of Science and Information Technology, The University of Newcastle, Callaghan, NSW 2308, Australia. Email: Raymond.Chiong@newcastle.edu.au

§Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey. Email: catay@sabanciuniv.edu

This is a preview version of this article [1] (see page 3 for the reference). It is posted here for your personal use and not for redistribution. The final publication and definitive version is available from IEEE (who hold the copyright) at https://www.ieee.org/. See also http://dx.doi.org/10.1109/SSCI.2015.242.

Abstract—The Vehicle Routing Problem with Time Windows (VRPTW) is a well-known combinatorial optimization problem found in many practical logistics planning operations. While exact methods designed for solving the VRPTW aim at minimizing the total distance traveled by the vehicles, heuristic methods usually employ a hierarchical objective approach in which the primary objective is to reduce the number of vehicles needed to serve the customers while the secondary objective is to minimize the total distance. In this paper, we apply a holistic approach that optimizes both objectives simultaneously. We consider several state-of-the-art Ant Colony Optimization (ACO) techniques from the literature, including the Min-Max Ant System, Ant Colony System, and Population-based Ant Colony Optimization (PACO). Our experimental investigation shows that PACO outperforms the others. Subsequently, we introduce a new pheromone matrix initialization approach for PACO (PI-PACO) that uses information extracted from the problem instance at hand and enforces pheromone assignments to edges that form feasible building blocks of tours. Our computational tests show that PI-PACO performs better than PACO. To further enhance its performance, we hybridize it with a local search method. The resulting algorithm is efficient in producing high quality solutions and outperforms similar hybrid ACO techniques.

I. INTRODUCTION

The Vehicle Routing Problem with Time Windows (VRPTW) is a well-known distribution logistics problem where a homogeneous fleet of m vehicles serves n geographically dispersed customers. Each customer ci has a demand wi and is associated with a service time si. Each customer has a time window [ei, li] during which servicing is allowed. ei is the earliest and li is the latest service time. The time window is assumed to be strict. In other words, the actual service start time bi at customer ci must satisfy ei ≤ bi ≤ li. So, any vehicle arriving at ci before ei must wait until ei. The vehicles reside in a central depot are denoted as 0 and associated with time window [e0, l0]. This time window implies that any vehicle must leave the depot after e0 and must return to the depot before l0. Each route originates and terminates at the depot. Each customer must be serviced exactly once by exactly one vehicle, and the total demand of the customers assigned to a route must not exceed the vehicle capacity.

The VRPTW has been extensively studied in the literature over the past three decades, with numerous exact and metaheuristic methods developed to solve it efficiently. While exact methods minimize the total distance traveled by the vehicles, metaheuristics usually optimize these objectives separately, adopting a hierarchical objective approach: the primary objective is to minimize the number of vehicles (f1) needed to serve the customers and the secondary objective is to minimize the total travel distance (f2). Different from heuristics, metaheuristics build and refine several solutions but they cannot guarantee optimality [2, 3]. Genetic Algorithms [4, 5], Ant Colony Optimization (ACO) [6], Simulated Annealing [7, 8], Tabu Search (TS) [9, 10, 11], Adaptive Large Neighborhood Search [12], Variable Neighborhood Search [13] as well as hybrid methods [14, 15] are among the metaheuristic approaches that have been successfully applied to the VRPTW. We refer the interested reader to [16] for an extensive review of metaheuristics applied to the VRPTW.

Many researchers have applied two-stage methods to solve the two objectives of the VRPTW separately, and metaheuristics are hybridized in each stage. Such two-stage hybridized algorithms are regarded as the most powerful methods to solve the VRPTW. Homberger and Gehring [14], for instance, chose an evolution strategy to generate solutions with the goal of reducing the number of vehicles in the first stage. Then, in the second stage, TS was used to minimize the total distance. Liu and Shen [17] introduced a two-stage metaheuristic based on a new neighborhood structure utilizing the relationship between routes and nodes. Developing such approaches requires carefully dividing the runtime between the local and
global parts of the hybrid algorithms as well as between the optimization of the two objectives. Different from these methods, we argue that optimizing the two objectives at the same time but in a prioritized way can lead to better results. It is also the more elegant and holistic approach. We investigate four ACO algorithms for this purpose: the Min-Max Ant System (MMAS) [18], the Ant Colony System (ACS) [19], our previously developed Initialized ACO (IACO) [20], and the Population-based ACO (PACO) algorithm [21, 22].

Note that distance minimization may arise as the only objective in scenarios where a company outsources its distribution/collection operations to third-party logistics service providers and is charged on a per-kilometer basis. ACO is shown to provide competitive results for this case too [23, 24].

The rest of the paper is structured as follows. In Section II, we describe some preliminaries for implementing the ACO algorithms. We then discuss the experimental setup and compare the results of four ACO variants in Section III. We present the probability initialization method and investigate its effects in Section IV. In Section V, we enhance the initialization of the pheromone trails and hybridize the algorithm with local search to further improve the solution quality. The performance of the proposed algorithm is compared with those of two state-of-the-art ACO-based approaches in Section VI. Finally, Section VII concludes the paper.

II. PRELIMINARIES

A. Solution Construction

A solution to the VRPTW is a schedule specifying which customers each vehicle serves and the timing of the service. The vehicle must service each of its customers within the time window without exceeding its capacity. The distance between two customers $c_i$ and $c_j$ is denoted as $d_{ij}$. In most of the benchmark problem instances (e.g., well-known instances generated by Solomon [25]), this is the Euclidean distance of customer coordinates in a two-dimensional plane. The travel time $t_{ij}$ between $c_i$ and $c_j$ is usually assumed to be equal to $d_{ij}$ and also represents the travel cost.

A permutation of customers $\pi = (c_{i_1}, c_{i_2}, \ldots)$ can be used to encode a route. All vehicles leave the depot at $e_0$. Let a vehicle first visit $c_i$ in the permutation $\pi$. The service start time $b_i$ at $c_i$ is then equal to $\max \{e_i, e_0 + t_{0i}\}$. The service at the second customer in the sequence $c_j$ will then start at time $b_j = \max \{e_j, b_i + s_i + t_{ij}\}$, where $s_i$ is the travel time between the customers visited or no other customer can be visited because of its time-window restriction. Then, the vehicle will return to the depot. This procedure is repeated with a new vehicle until all customers have been visited.

B. Objectives

The VRPTW has two objective functions: the number of vehicles needed to service all the customers $f_1$ and the total distance traveled by the vehicles $f_2$. These objectives do not have the same priority, i.e., minimizing $f_1$ is more important than $f_2$ (20), because it is assumed that the number of vehicles has more weight on total operating costs. In this study, we optimize both objectives simultaneously while respecting this prioritization. A solution $\pi_1$ is considered to be better than another solution $\pi_2$ if either $f_1(\pi_1) < f_1(\pi_2)$ or $f_1(\pi_1) = f_1(\pi_2)$ and $f_2(\pi_1) < f_2(\pi_2)$.

C. Pre-Processing

Some customer visiting sequences can be ruled out from any valid schedule, e.g., those that violate the time window constraints. By applying a pre-processing step [20], the search space can be reduced:

\[ c_j \text{ can be visited after } c_i \Rightarrow t_j \geq e_i + s_i + t_{ij} \quad (1) \]

Eq. 1 is a necessary condition to enable $c_j$ to be reached within the time window after servicing $c_i$. For each customer, we can determine the domain $[20]$, i.e., the set of customers that can be visited next in the same route. To select the customer to be visited next, we need to consider customers that are included in the current domain, and that the vehicle capacity limitation is not violated. The size of the search space can be reduced by up to 28% on average by this pre-processing procedure.

III. PERFORMANCE EVALUATION OF ACO ALGORITHMS

Before presenting an enhanced pheromone initialization approach, we investigated the performances of several ACO approaches using instances of the VRPTW. Specifically, we applied six algorithms to the Solomon [25] benchmark instances: MMAS, ACS, IACO, age-based PACO (PACO-ABS), quality-based PACO (PACO-QBS), and PACO with an elitist archive update strategy (PACO-EBS). When comparing two solutions during the optimization process, we first evaluated the numbers of vehicles, and if they were equal, we compared the total distances [20].

The Solomon data set consists of 25-, 50-, and 100-customer instances. The data includes 56 problems for each size, i.e., there are a total of 168 problems. We performed 20 independent runs for each of the settings. In each run, we granted 300,000 objective function evaluations (FEs). 20 ants were used to construct solutions. For all algorithms, we set the ACO parameters $\rho = 0.95$, $\beta = 2$, and $\alpha = 2$. Parameter values for IACO were set the same as in [20]. For the MMAS, $\tau_{\text{max}} = \frac{n}{f_2(\pi_{bs})}$ and $\tau_{\text{min}} = \frac{1}{e^*}$, where $f_2(\pi_{bs})$ is the travel distance of the best-so-far solution and $e^*$ is the average distance between the customers. For the ACS, $q_0 = 0.9$ and the local and global updating rules were defined according to [19]. For PACO, we set the archive size $K = 30$ and maximum pheromone $\tau_{\text{max}} = 60$.

We collected the results and used the Mann-Whitney U test with a significance level of 0.02 to check whether the observed differences in performance were significant. We tested the two objectives separately in order to identify in which objective an algorithm performs better. The final results are listed in Table I.

The columns marked with symbol $-$ (+, 0) show the number of instances “Algorithm 1” produces significantly better (worse, not significantly different) solutions than “Algorithm 2”. From Table I we can see that the ACS performs the worst for both $f_1$ and $f_2$. IACO performs worse than
the MMAS in small-size problems but better in large data instances: IACO has 35 smaller \( f_1 \) values than the MMAS out of 56 100-customer problems and is always better in terms of \( f_2 \). Overall, it outperforms the MMAS. The performance of PACO is slightly better than IACO in small-scale problems but significantly better in the 50- and 100-customer instances. In short, PACO performs the best among all tested ACO algorithms. We observe that PACO-ABS, PACO-QBS, and PACO-EMS show similar performance based on the 25-customer problem instances but PACO-QBS outperforms the other two when the larger data instances were used.

In the next section, we present a new pheromone initialization approach to improve the solution quality of PACO. Inspired by our previous work on IACO [28], the proposed approach is called probability initialization in view of the features of PACO.

IV. PROBABILITY-INITIALIZED PACO FOR THE VRPTW

In PACO, the probability to traverse a given edge is defined by two factors: (i) the number of solutions that contain the given edge in the archive and (ii) constant lower and upper pheromone limits (\( \tau^0_{ij} \) and \( \tau_{max} \)). During the course of PACO, solutions enter the archive and the transition probabilities are updated. We speed up this process by initializing the pheromone trails with knowledge extracted from the problem instance. We compute good per-edge lower pheromone bounds \( \tau_{ij}^0 \), i.e., the minimum pheromone values may now differ from edge to edge. High \( \tau_{ij} \) values indicate that it may be a good idea to visit customer \( c_j \) after \( c_i \). Our initialization process uses two components to determine such relationships: time windows and distance.

A. Initializing Pheromones in PACO

The service begin time \( b_i \) of a customer \( c_i \) is not known in advance, as it is the result of the vehicle schedule. This uncertainty propagates to subsequent customers. We, therefore, model \( b_i \) as a random variable \( x \) that complies with a certain probability distribution function denoted as \( PD_{i}(x) \). Different families of probability distributions investigated include normal, uniform, and power function distributions.

For the normal distribution, we set parameters \( \mu_i \) and \( \sigma_i \) as follows:

\[
P_D(x) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}
\]  

(2)

\[
\mu_i = \frac{e_i + l_i}{2}
\]  

(3)

\[
\sigma_i = \frac{l_i - e_i}{6}
\]  

(4)

For the uniform distribution, \( PD_i(x) \) is defined as follows:

\[
P_D(x) = \begin{cases} 
\frac{1}{l_i - e_i} & \text{if } e_i \leq x \leq l_i \\
0 & \text{otherwise}
\end{cases}
\]  

(5)

For the power function distribution, we set \( PD_i(x) \) as follows:

\[
P_D(x) = \begin{cases} 
\frac{1}{\ln(\frac{l_i}{e_i})(x+1)} & \text{if } e_i \leq x \leq l_i \\
0 & \text{otherwise}
\end{cases}
\]  

(6)

Assuming that customer \( c_j \) is near customer \( c_i \) and its latest service time is such that a vehicle departing from \( c_i \) may barely arrive at \( c_j \) before the latest service time. Then, it may make sense to increase the probability of visiting \( c_j \) directly after \( c_i \), since it will be hardly possible to insert another customer between \( c_i \) and \( c_j \). Using this idea, we derive the function \( VE(i, j, b_{i,m}) \) whose value is larger when customers \( c_i \) and \( c_j \) are closer and when \( c_j \) is serviced at the end of its time window if it is visited immediately after \( c_i \). \( VE(i, j, b_{i,m}) \) is formulated as follows:

\[
VE(i, j, b_{i,m}) = \begin{cases} 
0 & \text{if } b_j \geq l_j \\
\frac{\overline{d}_{ij}}{d_{ij}} + \gamma \left( \frac{\overline{d}_{ij}}{d_{ij}} - \frac{e_i - b_i}{b_j} \right) & \text{if } e_j > b_j \\
\frac{\overline{d}_{ij}}{d_{ij}} + \gamma \left( b_j - e_i \right) & \text{if } e_j \leq b_j < l_j
\end{cases}
\]  

(7)

where \( \overline{d}_{ij} \) is the average distance from customer \( c_i \) to all customers in its domain. \( l_j \) is the total length of the time window of customer \( c_j \). Under the assumption that customer \( c_i \) is visited before \( c_j \) and is serviced at \( b_{i,m} \), then \( b_j \) can be calculated as \( b_j = b_{i,m} + s_i + d_{ij} \). \( \frac{\overline{d}_{ij}}{d_{ij}} \) is used to give higher probabilities to closer customers and it decreases with \( d_{ij} \). \( \frac{e_i - b_i}{b_j} \) represents a penalty to punish early arrival. \( l_j - b_j + 1 \) is the time left to meet the latest service time. \( \gamma \) is a control
parameter governing the relative influence between distance and time. We set it to 1 in our initial experiments.

Eq. [7] considers three possible visits to a customer: failing to service due to late arrival; early arrival and waiting to service; and arrival within the time window and service immediately. In order to indicate the price and penalty for each type, we add or subtract a ratio. If a vehicle visits customer \( c_j \) immediately after customer \( c_i \), we can separate the time window of customer \( c_i \) into two disjoint parts: the waiting time \( WT_{i,j} \) and the active time \( VT_{i,j} \) during which service takes place. If the vehicle arrives during \( WT_{i,j} \) at \( c_j \), it has to wait and \( VE(i,j,b_{i,m}) = \frac{d_{i,j}}{d_{i,j}} + \gamma \left( \frac{t_{j}-t_{i}}{t_{j}-t_{i}} - \frac{t_{j}-b_{j}}{t_{j}-t_{i}} \right) \). If it arrives during \( VT_{i,j} \), it can immediately begin servicing \( c_j \) and \( VE(i,j,b_{i,m}) = \frac{d_{i,j}}{d_{i,j}} + \gamma \left( \frac{t_{j}-b_{j}+1}{t_{j}-b_{j}} \right) \).

To numerically estimate the initial pheromone \( \tau_{0,i,j} \) on the edge from \( c_i \) to \( c_j \) based on these situations, we divide the time window of \( c_i \) into \( M \) intervals of length \( \Delta b_i \), each associated with a service begin time \( b_{i,m} \). The computation is as follows:

\[
\tau_{0,i,j} = \max \left\{ \frac{1}{n} \sum_{m=1}^{M} PD_i(b_{i,m}) \ast VE(i,j,b_{i,m}) \ast \Delta b_i \right\}
\]  

(8)

B. Experimental Results

We refer to PACO equipped with the probability initialization described above as PI-PACO. To test its performance, we compared the results obtained with PI-PACO using the normal, uniform, and power distributions to the results achieved by PACO without the proposed initialization method. In Section III, we have observed that PACO-QBS outperforms PACO-ABS and PACO-EBS. Thus, in the subsequent experiments we used only PACO-QBS and it will be referred to simply as PACO. We again applied the Mann-Whitney U test at significance level 0.02 to check the final results. The parameters were set the same as in Section III.

The columns \(-, +, 0\) in Table III have the same meaning as in Section III. From the table, we can see that PACO initialized based on the power or uniform distribution performs significantly better than PACO in both \( f_1 \) and \( f_2 \). This means our initialization procedure improves the solution quality. Among the three choices of probability distributions, the normal distribution performs worse than the other two while the power distribution has a slight advantage over the uniform distribution, especially for larger instances. The results indicate that the proposed probability initialization approach is clearly beneficial and our algorithm is relatively robust in terms of the choice of initialization method.

V. INITIALIZATION ENHANCEMENT AND LOCAL SEARCH

A. Modification in the Initialization Approach

Pheromone initialization is a way to provide PACO with \textit{a priori} information about the problem in the form of (modified) transition probabilities from one customer to another. In an ideal case, edges whose pheromone values have been increased would form continuous sub-tours. In Figure 1 we plot the edges with the strongest initial pheromones for three of the benchmark problems.

We can see that this intended behavior occurs for clustered problems (C-type), but diminishes when the problem type changes to random/clustered mix (RC-type) or fully random (R-type) instances. The reason why rather distant and seemingly unrelated customers are linked in problem R101 could be that the relative influence between distance and time is not set very well at \( \gamma = 1 \). Thus, we decrease the value of \( \gamma \) to add weight to the distance between nodes. In order to determine the proper value of \( \gamma \), we take problem R101 as an example and display the same plots for \( \gamma \in \{0.1, 0.5, 1, 0.1, 1.5\} \) in Figure 2.

From the figure, we see that the lower the \( \gamma \) value, the nearer customers are linked by the increased pheromone values and the initialization seems to be able to pre-construct more useful solution fragments. In other words, lower values of \( \gamma \) (such as 0.1) appear to perform well.

In Figures 1 and 2 we can only spot a few solution fragments. Sometimes, several edges with strong initialized pheromones lead to the same customer. This is not desirable since each customer can only be visited once. The performance of the current initialization method may be improved, especially for random problem instances (R-type). These benchmark instances are characterized by random customer positions and random earliest service times. As a result, there are many potential choices for the next customer to visit following a given customer. To speed up the search by hinting PACO towards good solution building blocks, we want to reduce situations where multiple edges with strong initialized pheromone emanate from the same customer. For that, we decrease the initial pheromone values on certain edges to the minimum value. Assume that the current customer is \( c_i \) and we want to only keep the initialized pheromone on one edge going into \( c_i \). We tested two ways to choose this edge: First, we can keep the one with the largest value (maximum selection). However, this edge could come from a customer that has several edges with similarly high pheromone values leaving it, while the edge whose pheromone we delete could be the one with the highest pheromone value leaving its origin. The second approach is to compare all incoming edges of \( c_i \). If two such edges are those with the highest pheromone leaving their origins, we pick the one with the largest pheromone difference to the second highest pheromone on any edge leaving its origin (difference selection).

In Figure 3 we sketch the impact of the difference selection approach in problem instances C201, RC201, and R201 compared to the original initialization procedure. As expected, the pheromone values have increased on one departing edge per customer. Some customers become isolated due to weaker pheromones to customers in their domains. In other words, these customers show random distribution characteristics. Moreover, this figure illustrates the different patterns for different problem types. In C201, the increased pheromones almost outline complete tours. However, in RC201 and R201 with random features, only a few segments are linked. From these figures, we preliminarily conclude that the selection methods can extract not only \textit{a priori} information from problems but also maintain problem characteristics. The maximum selection approach (not illustrated) has a similar visual appearance as the difference selection approach. In Algorithm, we show
TABLE II: Mann-Whitney U test results for PACO with different strategies.

<table>
<thead>
<tr>
<th>Algorithm 1 vs. 2</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_1$</th>
<th>$f_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>NoIni</td>
<td>0</td>
<td>2</td>
<td>54</td>
<td>4</td>
<td>26</td>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td>Normal</td>
<td>Power</td>
<td>0</td>
<td>3</td>
<td>53</td>
<td>1</td>
<td>33</td>
<td>22</td>
<td>5</td>
</tr>
<tr>
<td>Normal</td>
<td>Uniform</td>
<td>0</td>
<td>2</td>
<td>54</td>
<td>1</td>
<td>39</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>Normal</td>
<td>Power</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>1</td>
<td>13</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>Normal</td>
<td>Uniform</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>0</td>
<td>19</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>Power</td>
<td>Uniform</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>4</td>
<td>0</td>
<td>52</td>
<td>0</td>
</tr>
</tbody>
</table>

(a) C101
(b) RC101
(c) R101

Fig. 1: Edges with the strongest initial pheromone values for three of the Solomon problems based on $\gamma = 1$.

(a) $\gamma = 0.1$
(b) $\gamma = 0.5$
(c) $\gamma = 1.0$
(d) $\gamma = 1.5$

Fig. 2: Edges with the strongest initial pheromone values for problem R101 with varied values of $\gamma$.

the complete process of the probability initialization method. This procedure is executed once at the beginning of PI-PACO.

B. Local Search

Local search algorithms are optimization methods that maintain and improve a candidate solution by exploiting on one (or multiple) neighborhood structure(s). They are widely used in solving the many VRP variants, often in a hybrid form combined with global search methods, e.g., further improving the iteration-best solution found by ACO. In this section, we hybridize PI-PACO with local search. We first introduce different neighborhood structures explored and then present the new hybrid algorithm.

Local search is an effective method to improve solutions obtained by metaheuristics algorithms and has been successfully employed with ACO as well, e.g., see [13, 23, 27]. Our local search method is based on three well-known neighborhood structures: $N_{1-insert}$, $N_{1-exchange}$, and $N_{2-opt}$. We assume two routes $\pi_1$ and $\pi_2$ ($\pi_1, \pi_2 \in \pi$). $N_{1-insert}$ defines an insertion move where a customer is removed from route $\pi_1$ and inserted into route $\pi_2$. Figure 4a depicts the example where customer $c_4$ is removed from route 1 and inserted between customer $c_3$ and customer $c_5$ in route 2. $N_{1-exchange}$ exchanges the positions of two customers within the same route or between two routes. In Figure 4b, $c_4$ replaces $c_5$ in route 1 and $c_5$ replaces $c_4$ in route 2. This method cannot reduce the number of routes but may reduce the total distance. $N_{2-opt}$ removes two arcs in $\pi_1$ and $\pi_2$ and reconnects the two resulting paths. In Figure 4c, the arc $(c_3, c_5)$ is eliminated and $c_4$ is reconnected to $c_0$ in route 1 whereas the arc $(c_4, c_5)$ in route 2 is removed and the arc $(c_4, c_5)$ is constructed.

These neighborhood structures may contain infeasible solutions. We therefore check the feasibility before applying an operator to certain customers to ensure that each generated solution is feasible.

It is difficult to decide which search operator to use when we
do not know how the solutions are composed. Thus, in many algorithms, more than one search operator would be applied. Some have been used in different phases of the algorithms, e.g., see [15, 27].

Homberger and Gehring [14] proposed a hybrid meta-heuristic that randomly selects a neighborhood structure among \( \{ N_1-\text{insert}, N_1-\text{exchange}, N_2-\text{opt} \} \) and applies it. We have adopted the same mechanism in PI-PACO.

The pseudo-code of the local search is given in Algorithm 2. Figure 5 shows a flowchart of the complete hybrid PI-PACO.

VI. COMPUTATIONAL EVALUATION

We experimentally investigated the performance of the two selection approaches in conjunction with the hybridization of our algorithm with the local search method. In Table III we show the results obtained by PI-PACO (aggregated for each problem type) and two state-of-the-art ACO algorithms [15, 27]. We set \( K = 5 \), \( \tau_{\text{max}} = 90 \), and \( \gamma = 0.1 \).

Comparing the results obtained by the two modified pheromone initialization methods, we see that, in all six problem types, the maximum selection approach is worse than the difference selection approach that preserves the pheromone on the edges according to their difference to the next-strongest pheromone. We also notice that the hybrid PI-PACO with

<table>
<thead>
<tr>
<th>Type</th>
<th>Goal</th>
<th>Chen and Ting [15]</th>
<th>Sodsoon and Changyom [27]</th>
<th>Hybrid PI-PACO maximum selection</th>
<th>Hybrid PI-PACO difference selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>( f_1 )</td>
<td>12.83</td>
<td>13.83</td>
<td>12.83</td>
<td>12.92</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>1203.56</td>
<td>1259.19</td>
<td>1204.06</td>
<td>1205.11</td>
</tr>
<tr>
<td>C1</td>
<td>( f_1 )</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>828.76</td>
<td>838.12</td>
<td>828.61</td>
<td>828.60</td>
</tr>
<tr>
<td>RC1</td>
<td>( f_1 )</td>
<td>12.50</td>
<td>12.63</td>
<td>12.75</td>
<td>12.63</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>1363.84</td>
<td>1436.58</td>
<td>1381.42</td>
<td>1380.78</td>
</tr>
<tr>
<td>R2</td>
<td>( f_1 )</td>
<td>3.09</td>
<td>3.82</td>
<td>3.45</td>
<td>3.64</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>932.23</td>
<td>980.98</td>
<td>1005.35</td>
<td>995.03</td>
</tr>
<tr>
<td>C2</td>
<td>( f_1 )</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>589.86</td>
<td>591.13</td>
<td>590.71</td>
<td>589.93</td>
</tr>
<tr>
<td>RC2</td>
<td>( f_1 )</td>
<td>3.75</td>
<td>4.5</td>
<td>4.13</td>
<td>4.38</td>
</tr>
<tr>
<td></td>
<td>( f_2 )</td>
<td>1079.81</td>
<td>1141.63</td>
<td>1113.59</td>
<td>1156.20</td>
</tr>
</tbody>
</table>

**Table III: Comparisons between improved initialization method and ACO-based algorithms.**
difference selection achieves better results than hybrid PACO without pheromone initialization. This shows that the modification to our initialization method is useful and can improve the solution quality.

Our algorithm outperforms the hybrid algorithm by Chen and Ting [15] on problem type C1 and achieves similar results on problem type C2. This indicates that our method is good for clustered problems. In problem types R1 and RC1, our method achieves better values in the first objective function at the cost of slightly longer travel distance. In problem types R2 and RC2, however, our algorithm needs half a vehicle more on average. These results show that our method is weaker in solving problems with wide time windows.

Our algorithm is similar to the MMAS-VRPTW [27] but outperforms it on all except R2 instances in terms of the distance, and scores equally in C1 instances in terms of the number of vehicles. In summary, our algorithm has used a total of 438.02 vehicles on average compared to 459.02 vehicles used by the MMAS-VRPTW, and outperformed two state-of-the-art ACO methods on the VRPTW.

VII. Conclusions

In this paper, we have addressed the VRPTW using an ACO approach. Our contributions can be summarized as follows:

- We showed that pure PACO outperforms other ACO algorithms on the VRPTW.
- We introduced a method to initialize the pheromone of PACO and showed that this method outperforms pure PACO.
- We improved this initialization procedure so that it could create more useful tour fragments.
- We hybridized PACO with local search using three different search operators.

Algorithm 1 Pheromone Initialization

1: for $c_i$, $i = 1...n$ do
2:   \[
    PD_i(x) = \begin{cases} 
    \frac{1}{n} & \text{if } e_i \leq x \leq l_i \\
    0 & \text{otherwise}
    \end{cases}
\] (9)
3: for $c_j$, $j = 1...n$ do
4:   \[
    VE(i,j,b_{i,m}) = \begin{cases} 
    \frac{h_j}{h_j + \gamma \left( \frac{c_j-b_j}{l_j} \right)} & \text{if } b_j \geq l_j \\
    \frac{h_j}{h_j + \gamma \left( \frac{c_j-b_j}{l_j} \right)} & \text{if } c_j > b_j \\
    \frac{h_j}{h_j + \gamma \left( \frac{c_j-b_j}{l_j} \right)} & \text{if } e_j \leq b_j < l_j
    \end{cases}
\] (10)
5:   \[
    \tau_{i,j} = \sum_{m=1}^{M} PD_i(b_{i,m}) \ast VE(i,j,b_{i,m}) + \Delta b_i
\] (11)
6: \[
    \tau_{i,j}^0 = \max \{ \frac{1}{n}, \tau_{i,j} \} 
\] (12)
7: end for
8: end for
9: for $c_i$, $i = 1...n$ do
10:   choose a. or b.
11:   a. maximum selection of pheromone value
12:      \[
      j = \arg \max \{ \tau_{k,i} | c_i \text{ is in the domain of } c_k \} 
\] (13)
13:      \[
      \tau_{k,i}^0 = \begin{cases} 
      \frac{1}{n} & k \neq j \\
      \tau_{k,i} & k = j \text{ (unchanged)}
      \end{cases}
\] (14)
14:   b. difference selection of pheromone value
15:      \[
      j = \arg \max \{ \tau_{k,i} | c_k \text{ is in domain of } c_j \} 
\] (15)
16:      \[
      p = \arg \max \{ \tau_{p,k} | c_k \text{ is in domain of } c_p \} 
\] (16)
17:      \[
      x = \arg \max \{ \tau_{k,r} | (c_k \text{ is in domain of } c_j) \land (x \neq j) \} 
\] (17)
18:      \[
      y = \arg \max \{ \tau_{k,p} | (c_k \text{ is in domain of } c_p) \land (y \neq p) \} 
\] (18)
19:      \[
      \tau_{j,i}^0 = \begin{cases} 
      \tau_{k,i} & (\tau_{j,i} - \tau_{j,z} \geq \tau_{p,j} - \tau_{p,y} \text{ (unchanged)} \\
      \tau_{k,i}^0 & \text{otherwise}
      \end{cases}
\] (19)
20: end for

Algorithm 2 The description of the local search.

Input: $\pi^{ib}$

Output: $\pi^{ib}$

1: $N^* = \text{random}\{N_{1\text{-insert}}, N_{1\text{-exchange}}, N_{2\text{-opt}}\}$;
2: Generate $N^*$ of $\pi^{ib}$;
3: for $\pi$ in $N^*$ do
4:   if $(f_1(\pi) < f_1(\pi^{ib}))$ or $[(f_1(\pi) = f_1(\pi^{ib})) \land (f_2(\pi) < f_2(\pi^{ib}))]$ then
5:      $\pi^{ib} = \pi$;
6:      $\pi^{ib}$ replaces $\pi$ in $A$;
7:   end if
8: end for
9: $\pi^{ib} = \pi^{ib}$;
We showed that this hybrid PACO with improved initialization outperforms state-of-the-art hybrid algorithms and pure PACO.

Future research may focus on applying the proposed PL-PACO approach to other problems, e.g., the Quadratic Assignment Problem. Alternative local search algorithms may also be explored to determine the best hybridization strategy. Furthermore, the performance of PL-PACO may be investigated on large-scale instances. In this case, the effectiveness of the probability initialization method needs to be analyzed and extraction of useful a priori information from a large instance needs to be explored.

Acknowledgements. We acknowledge support from the Fundamental Research Funds for the Central Universities, the National Natural Science Foundation of China under Grant 61150110488, and the European Union 7th Framework Program under Grant 247619.

REFERENCES


This is a preview version of this article [1] (see page 9 for the reference). It is posted here for your personal use and not for redistribution. The final publication and definite version is available from IEEE (who hold the copyright) at https://www.ieee.org/. See also http://dx.doi.org/10.1109/SSCI.2015.242