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

***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 c_i has a demand w_i and is associated with a service time s_i. Each customer has a time window $[e_i, l_i]$ during which servicing is allowed. e_i is the earliest and l_i is the latest service time. The time window is assumed to be strict. In other words, the actual service start time b_i at customer c_i must satisfy $e_i \leq b_i \leq l_i$. So, any vehicle arriving at c_i before e_i must wait until e_i. The vehicles reside in a central depot are denoted as 0 and associated with time window $[e_0, l_0]$. This time window implies that any vehicle must leave the depot after e_0 and must return to the depot before l_0. 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 (f_1) needed to serve the customers and the secondary objective is to minimize the total travel distance (f_2). Different from heuristics, metaheuristics build and refine several solutions but they cannot guarantee optimality [1, 2]. Genetic Algorithms [3, 4], Ant Colony Optimization (ACO) [5], Simulated Annealing [6, 7], Tabu Search (TS) [8, 9, 10], Adaptive Large Neighborhood Search [11], Variable Neighborhood Search [12] as well as hybrid methods [13, 14] are among the metaheuristic approaches that have been successfully applied to the VRPTW. We refer the interested reader to [15] 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 [13], 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 [16] 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

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.242

System (MMAS) [17], the Ant Colony System (ACS) [18], our previously developed Initialized ACO (IACO) [19], and the Population-based ACO (PACO) algorithm [20, 21].

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 [22, 23].

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 [24]), 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 $\pi = (c_i, c_j, \dots)$ 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 π. 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 $b_j = \max\{e_j, b_i + s_i + t_{ij}\}$. This is repeated for the remaining customers in the schedule until either the vehicle capacity is reached by the aggregated demands of 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 ([25]), 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 π_1 is considered to be better than another solution π_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 [19], the search space can be reduced:

$$c_j \text{ can be visited after } c_i \Rightarrow l_j \geq e_i + s_i + t_{ij} \qquad (1)$$

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* [19], 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 [24] 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 [19].

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 [19]. For the MMAS, $\tau_{max} = \frac{n}{f_2(\pi_{bs})}$ and $\tau_{min} = \frac{1}{c^* n^2}$, where $f_2(\pi_{bs})$ is the travel distance of the best-so-far solution and c^* 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 [18]. For PACO, we set the archive size $K = 30$ and maximum pheromone $\tau_{max} = 60$.

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.

TABLE I: Mann-Whitey U test comparison results for ACO algorithms ($-$ is better, $+$ is worse).

		Instances with 25 customers						Instances with 50 customers						Instances with 100 customers						Total					
		f_1			f_2			f_1			f_2			f_1			f_2			f_1			f_2		
Algorithm 1 vs. 2		-	+	0	-	+	0	-	+	0	-	+	0	-	+	0	-	+	0	-	+	0	-	+	0
ACS	IACO	0	40	16	0	52	4	0	40	16	1	51	4	0	44	12	0	54	2	0	124	44	1	157	10
ACS	MMAS	0	49	7	1	51	4	0	41	15	9	36	11	2	36	18	35	7	14	2	126	40	45	94	29
ACS	PACO-ABS	0	12	44	9	31	16	0	35	21	5	45	6	0	54	2	0	51	5	0	101	67	14	127	27
ACS	PACO-EBS	0	47	9	0	54	2	0	48	8	0	54	2	0	55	1	0	52	4	0	150	18	0	160	8
ACS	PACO-QBS	0	45	11	0	55	1	0	49	7	0	55	1	0	56	0	0	56	0	0	150	18	0	166	2
IACO	MMAS	0	17	39	13	28	15	9	7	40	38	7	11	35	1	20	56	0	0	44	25	99	107	35	26
IACO	PACO-ABS	0	12	44	9	31	16	0	35	21	5	45	6	0	54	2	0	51	5	0	101	67	14	127	27
IACO	PACO-EBS	0	9	47	9	28	19	0	31	25	4	43	9	0	42	14	2	39	15	0	82	86	15	110	43
IACO	PACO-QBS	0	13	43	8	34	14	0	37	19	4	49	3	0	54	2	0	52	4	0	104	64	12	135	21
MMAS	PACO-ABS	5	1	50	15	12	29	0	31	25	1	49	6	0	54	2	0	56	0	5	86	77	16	117	35
MMAS	PACO-EBS	3	1	52	19	12	25	0	29	27	0	49	7	0	53	3	0	56	0	3	89	82	19	117	3 2
MMAS	PACO-QBS	3	1	52	16	15	25	0	33	23	0	52	4	0	55	1	0	56	0	3	83	76	16	123	29
PACO-ABS	PACO-EBS	0	0	56	5	1	50	3	0	53	19	2	35	28	0	28	48	0	8	31	0	137	72	3	93
PACOABS	PACO-QBS	0	0	56	0	3	53	0	1	55	0	13	43	0	1	55	1	28	27	0	2	166	1	44	123
PACO-EBS	PACO-QBS	1	0	55	0	18	38	0	4	52	0	34	22	0	39	17	0	55	1	1	43	124	0	107	61

In short, PACO performs the best among all tested ACO algorithms. We observe that PACO-ABS, PACO-QBS, and PACO-EBS show similar performances 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 [19], 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 (τ_{ij}^0 and τ_{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 τ_{ij}^0, i.e., the minimum pheromone values may now differ from edge to edge. High τ_{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 μ_i and σ_i as follows:

$$PD_i(x) = \frac{1}{\sigma_i\sqrt{2\pi}}e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \quad (2)$$

$$\mu_i = \frac{e_i + l_i}{2} \quad (3)$$

$$\sigma_i = \frac{l_i - e_i}{6} \quad (4)$$

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

$$PD_i(x) = \begin{cases} \frac{1}{l_i - e_i} & \text{if } e_i \le x \le l_i \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

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

$$PD_i(x) = \begin{cases} \frac{1}{In(\frac{l_i+1}{e_i+1})(x+1)} & \text{if } e_i \le x \le l_i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

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:

$$VE(i, j, b_{i,m}) = \begin{cases} 0 & \text{if } b_j \ge l_j \\ \frac{\overline{d_i}}{d_{ij}} + \gamma\left(\frac{\overline{t_j}}{\overline{t_j}+1} - \frac{e_j - b_j}{\overline{t_j}}\right) & \text{if } e_j > b_j \\ \frac{\overline{d_i}}{d_{ij}} + \gamma\frac{\overline{t_j}}{l_j - b_j + 1} & \text{if } e_j \le b_j < l_j \end{cases} \quad (7)$$

where $\overline{d_i}$ is the average distance from customer c_i to all customers in its domain. $\overline{t_j}$ 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 $b_j = b_{i,m} + s_i + d_{ij}$. $\frac{\overline{d_i}}{d_{ij}}$ is used to give higher probabilities to closer customers and it decreases with d_{ij}. $\frac{e_j - b_j}{\overline{t_j}}$ represents a penalty to punish early arrival. $l_j - b_j + 1$ is the time left to meet the latest service time. γ 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 $VE(i, j, b_{i,m}) = \frac{\overline{d_i}}{d_{i\,j}} + \gamma(\frac{\overline{t_j}}{\overline{t_j}+1} - \frac{e_j - b_j}{\overline{t_j}})$. If it arrives during $VT_{i,j}$, it can immediately begin servicing c_j and $VE(i, j, b_{i,m}) = \frac{\overline{d_i}}{d_{i\,j}} + \gamma\frac{\overline{t_j}}{l_j - b_j + 1}$.

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 Δb_i, each associated with a service begin time $b_{i,m}$. The computation is as follows:

$$\tau^0_{i\,j} = \max\left\{\frac{1}{n}, \sum_{m=1}^{M} PD_i(b_{i,m}) * VE(i, j, b_{i,m}) * \Delta b_i\right\} \quad (8)$$

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 $-$, $+$, and 0 in Table II have the same meaning as in Section III. From the table, we can see that PACO initilized 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 *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.

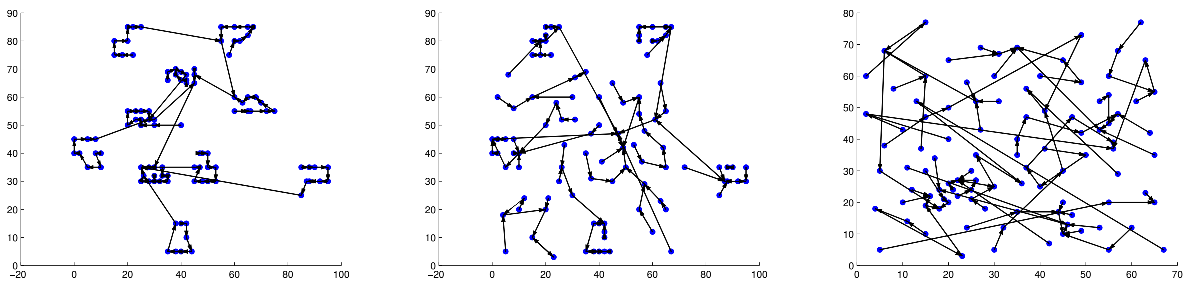
We can see that this intended behavior occurs for clustered problems (C-type), but diminishes when the problem type changes to random/clustered mix (RC-type) or fully random (R-type) instances. The reason why rather distant and seemingly unrelated customers are linked in problem R101 could be that the relative influence between distance and time is not set very well at $\gamma = 1$. Thus, we decrease the value of γ to add weight to the distance between nodes. In order to determine the proper value of γ, we take problem R101 as an example and display the same plots for $\gamma \in \{0.1, 0.5, 1.0, 1.5\}$) in Figure 2. From the figure, we see that the lower the γ 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 γ (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 *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. 1, we show the complete process of the probability initialization method. This procedure is executed once at the beginning of PI-PACO.

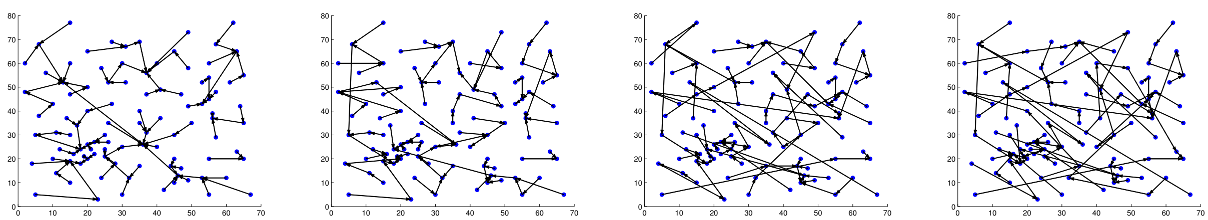
TABLE II: Mann-Whitney U test results for PACO with different strategies.

		Instances with 25 customers						Instances with 50 customers						Instances with 100 customers						Total					
		f_1			f_2			f_1			f_2			f_1			f_2			f_1			f_2		
Algorithm 1 vs. 2		-	+	0	-	+	0	-	+	0	-	+	0	-	+	0	-	+	0	-	+	0	-	+	0
NoIni	Normal	0	2	54	4	26	26	4	7	45	6	32	18	8	14	34	4	32	20	12	23	133	14	90	64
NoIni	Power	0	3	53	1	33	22	2	7	47	3	35	18	5	13	38	4	34	18	7	23	138	8	102	58
NoIni	Uniform	0	2	54	1	39	16	1	7	48	4	40	12	3	15	38	2	42	12	4	24	140	7	121	40
Normal	Power	0	0	56	1	13	42	0	2	54	1	8	47	0	1	55	1	6	49	0	3	165	3	27	138
Normal	Uniform	0	0	56	0	19	37	0	3	53	1	13	42	0	3	53	0	15	41	0	6	162	1	47	120
Power	Uniform	0	0	56	4	0	52	0	0	56	3	1	52	0	0	56	9	0	47	0	0	168	16	1	151



(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 γ.

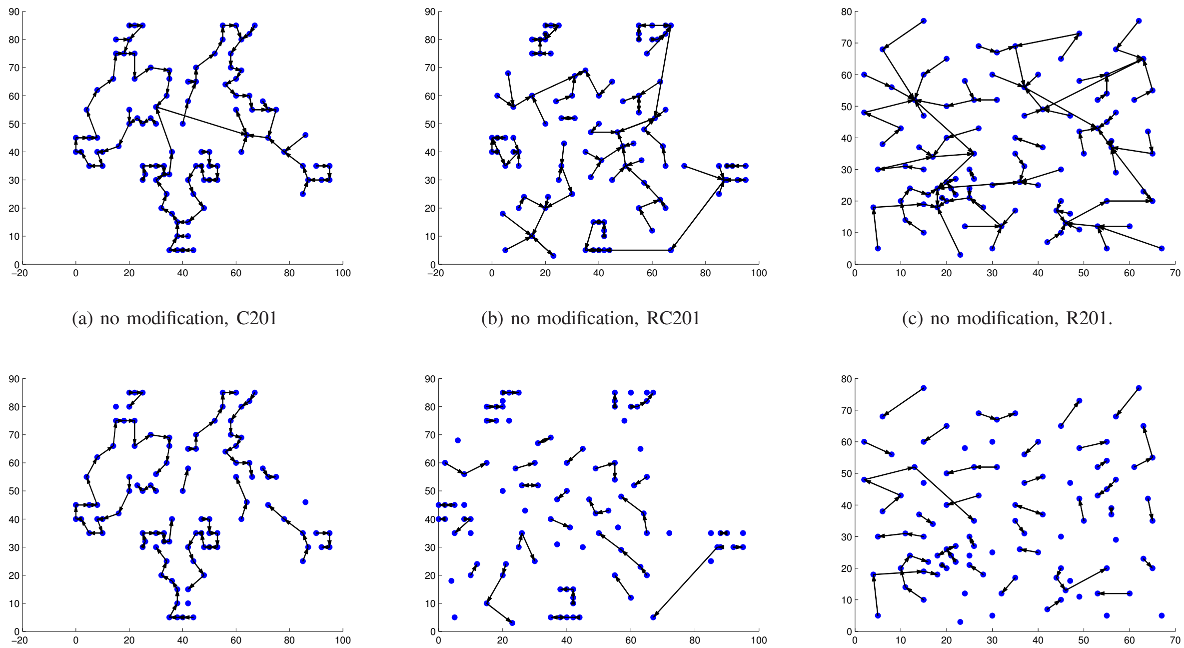
B. Local Search

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 [14, 22, 26]. 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 π_1 and π_2 ($\pi_1, \pi_2 \in \pi$). $N_{1-insert}$ defines an insertion move where a customer is removed from route π_1 and inserted into route π_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 π_1 and π_2 and reconnects the two resulting paths. In Figure 4c, the arc (c_3, c_5) is eliminated and c_3 is reconnected to c_0 in route 1 whereas the arc (c_4, c_0) 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.

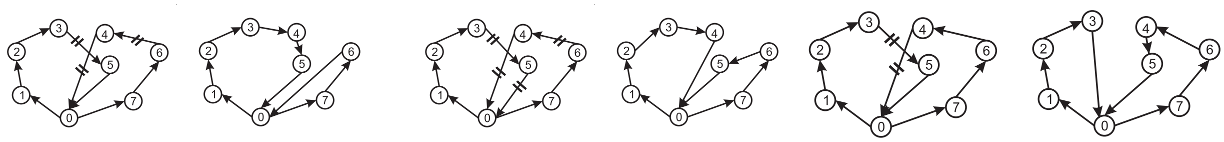
It is difficult to decide which search operator to use when we do not know how the solutions are composed. Thus, in many algorithms, more than one search operator would be applied. Some have been used in different phases of the algorithms,



(a) no modification, C201 (b) no modification, RC201 (c) no modification, R201.

(d) difference selection, C201. (e) difference selection, RC201. (f) bigger difference, R201.

Fig. 3: Difference selection for pruning initialized pheromone compared to the original initialized pheromone distribution.



(a) $N_{1-insert}$ (b) $N_{1-exchange}$ (c) N_{2-opt}

Fig. 4: The three neighborhood structures explored by the local search in our hybrid PACO.

e.g., see [14, 26].

Homberger and Gehring [13] proposed a hybrid metaheuristic that randomly selects a neighborhood structure among $\{N_{1-insert}, N_{1-exchange}, N_{2-opt}\}$ 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 [14, 26]. We set $K = 5$, $\tau_{max} = 90$, 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 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

TABLE III: Comparisons between improved initialization method and ACO-based algorithms.

Type	Goal	Chen and Ting [14]	Sodsoon and Changyom [26]	hybrid PACO	hybrid PI-PACO maximum selection	hybrid PI-PACO difference selection
R1	f_1	12.83	13.83	12.83	12.92	**12.75**
	f_2	1203.56	1259.19	1204.06	1205.11	1203.67
C1	f_1	**10**	**10**	**10**	**10**	**10**
	f_2	828.76	838.12	828.61	828.60	**828.55**
RC1	f_1	12.50	12.63	12.75	12.63	**12.38**
	f_2	1363.84	1436.58	1381.42	1380.78	1380.54
R2	f_1	**3.09**	3.82	3.45	3.64	3.54
	f_2	**932.23**	980.98	1005.35	995.03	1006.38
C2	f_1	**3**	**3**	3	**3**	**3**
	f_2	**589.86**	591.13	590.71	589.93	**589.86**
RC2	f_1	**3.75**	4.5	4.13	4.38	4.13
	f_2	**1079.81**	1141.63	1113.59	1156.20	1109.8

Algorithm 1 Pheromone Initialization

for c_i, $i = 1...n$ **do**

$$PD_i(x) = \begin{cases} \frac{1}{In(\frac{l_i+1}{e_i+1})(x+1)} & \text{if } e_i \le x \le l_i \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

  for c_j, $j = 1...n$ **do**

$$VE(i,j,b_{i,m}) = \begin{cases} 0 & \text{if } b_j \ge l_j \\ \frac{\overline{d_i}}{d_{ij}} + \gamma\left(\frac{\overline{t_j}}{\overline{t_j}+1} - \frac{e_j - b_j}{\overline{t_j}}\right) & \text{if } e_j > b_j \\ \frac{\overline{d_i}}{d_{ij}} + \gamma\frac{\overline{t_j}}{l_j - b_j + 1} & \text{if } e_j \le b_j < l_j \end{cases} \quad (10)$$

$$\tau_{ij}^{'} = \sum_{m=1}^{M} PD_i(b_{i,m}) * VE(i,j,b_{i,m}) * \Delta b_i \quad (11)$$

$$\tau_{ij}^{0} = max\{\frac{1}{n}, \tau_{ij}^{'}\} \quad (12)$$

  end for
end for
for c_i, $i = 1...n$ **do**
  choose a. or b.
  a. maximum selection of pheromone value

$$j = \arg\max\{\tau_{ki} | c_i \text{ is in the domain of } c_k\} \quad (13)$$

$$\tau_{ki}^{0} = \begin{cases} \frac{1}{n} & k \ne j \\ \tau_{ki}^{0} & k = j \text{ (unchanged)} \end{cases} \quad (14)$$

  b. difference selection of pheromone value

$$j = \arg\max\{\tau_{jk} : c_k \in \text{domain of } c_j\} \quad (15)$$

$$p = \arg\max\{\tau_{pk} : c_k \in \text{domain of } c_p\} \quad (16)$$

$$x = \arg\max\{\tau_{xk} : (c_k \in \text{domain of } c_j) \wedge (x \ne j)\} \quad (17)$$

$$y = \arg\max\{\tau_{yk} : (c_k \in \text{domain of } c_p) \wedge (y \ne p)\} \quad (18)$$

$$\tau_{ji}^{0} = \begin{cases} \tau_{ki}^{0} & \text{if } \tau_{ji} - \tau_{jx} \ge \tau_{pj} - \tau_{py} \text{ (unchanged)} \\ \frac{1}{n} & \text{otherwise} \end{cases} \quad (19)$$

end for

Algorithm 2 The description of the local search.

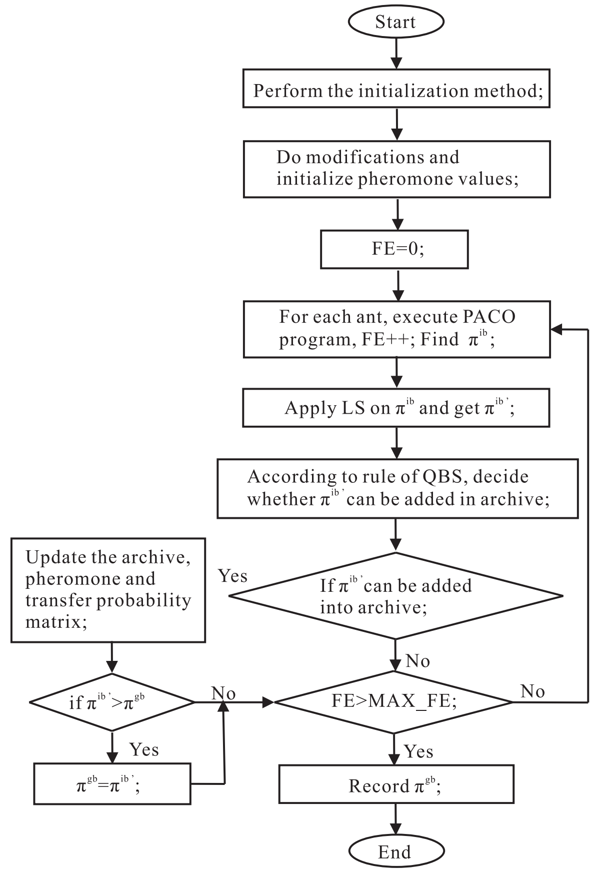
Input: π^{ib}
Output: $\pi^{ib'}$

N^*= random$\{N_{1-insert}, N_{1-exchange}, N_{2-opt}\}$;
Generate N^* of π^{ib};
for π in N^* **do**
  if $(f_1(\pi) < f_1(\pi^{ib})) \vee [(f_1(\pi) = f_1(\pi^{ib})) \wedge (f_2(\pi) < f_2(\pi^{ib}))]$ **then**
    $\pi^{ib} = \pi$;
    π^{ib} replaces $\pi^{'}$ in A;
  end if
end for
$\pi^{ib'} = \pi^{ib}$;

the solution quality.

Our algorithm outperforms the hybrid algorithm by Chen and Ting [14] 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.

Fig. 5: The flowchart of hybrid PI-PACO.



Our algorithm is similar to the MMAS-VRPTW [26] 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.
- 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 PI-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 PI-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

[1] C. Blum, R. Chiong, M. Clerc, K. De Jong, Z. Michalewicz, F. Neri, and T. Weise, "Evolutionary optimization," in *Variants of Evolutionary Algorithms for Real-World Applications*, R. Chiong, T. Weise, and Z. Michalewicz, Eds. Springer, 2012, pp. 1–29.

[2] R. Chiong, F. Neri, and R. I. McKay, "Nature that breeds solutions," in *Nature-Inspired Informatics for Intelligent Applications and Knowledge Discovery: Implications in Business, Science and Engineering*, R. Chiong, Ed. Information Science Reference, 2010, ch. 1, pp. 1–24.

[3] J. Berger and M. Barkaoui, "A parallel hybrid genetic algorithm for the vehicle routing problem with time windows," *Computers and Operations Research*, vol. 31, no. 12, pp. 2037–2053, 2004.

[4] J.-Y. Potvin and S. Bengio, "The vehicle routing problem with time windows part II: Genetic search," *INFORMS Journal on Computing*, vol. 8, no. 2, pp. 165–172, 1996.

[5] L. M. Gambardella, É. Taillard, and G. Agazzi, "MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London, U.K.: McGraw-Hill, 1999.

[6] H. Li and A. Lim, "Local search with annealing-like restarts to solve the VRPTW," *European Journal of Operational Research*, vol. 150, pp. 115–127, 2003.

[7] W.-C. Chiang and R. A. Russell, "Simulated annealing metaheuristics for the vehicle routing problem with time windows," *Annals of Operations Research*, vol. 63, no. 1, pp. 3–27, 1996.

[8] J. Homberger and H. Gehring, "Two evolutionary metaheuristics for the vehicle routing problem with time windows," *Information Systems and Operational Research*, vol. 37, pp. 297–318, 1999.

[9] P. Badeau, F. Guertin, M. Gendreau, J.-Y. Potvin, and E. Taillard, "A parallel tabu search heuristic for the vehicle routing problem with time windows," *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 2, pp. 109–122, 1997.

[10] R. A. Russell, "Hybrid heuristics for the vehicle routing problem with time windows," *Transportation Science*, vol. 29, no. 2, pp. 156–166, 1995.

[11] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers and Operations Research*, vol. 34, no. 8, pp. 2403–2435, 2007.

[12] L.-M. Rousseau, M. Gendreau, and G. Pesant, "Using constraint-based operators to solve the vehicle routing problem with time windows," *Journal of Heuristics*, vol. 8, no. 1, pp. 43–58, 2002.

[13] J. Homberger and H. Gehring, "A two-phase hybrid metaheuristic for the vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 162, pp. 220–238, 2005.

[14] C.-H. Chen and C.-J. Ting, "A hybrid ant colony system for vehicle routing problem with time windows," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 6, pp. 2822–2836, 2005.

[15] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, part II: Metaheuristics," *Transportation Science*, vol. 39, no. 1, pp. 119–139, 2005.

[16] F.-H. F. Liu and S.-Y. Shen, "A route-neighborhood-based metaheuristic for vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 118, pp. 485–504, 1999.

[17] T. Stützle and H. H. Hoos, "Max-min ant system," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.

[18] 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, 1997.

[19] W. Shi and T. Weise, "An initialized ACO for the VRPTW," in *Proceedings of the 14th International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2013, pp. 93–100.

[20] M. Guntsch and M. Middemdorf, "A population based approach for ACO," in *Applications of Evolutionary Computing*, ser. Lecture Notes in Computer Science, vol. 2279. Springer, 2002, pp. 72–81.

[21] S. Oliveira, M. S. Hussin, and T. Stützle, "A detailed analysis of the population-based ant colony optimization algorithm for the TSP and the QAP," Université Libre de Bruxelles, IRIDIA, Belgium, Tech. Rep. 006, 2011.

[22] U. M. Yıldırım and B. Çatay, "A time-based pheromone approach for the ant system," *Optimization Letters*, vol. 6, no. 6, pp. 1081–1099, 2012.

[23] U. M. Yıldırım and B. Çatay, "A parallel matheuristic for solving the vehicle routing problems," in *Computer-based Modelling and Optimization in Transportation*, J. F. de Sousa and R. Rossi, Eds. Springer International Publishing, 2014, vol. 262, pp. 477–489.

[24] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.

[25] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, Massachusetts: The MIT Press, 2007.

[26] S. Sodsoon and P. Changyom, "Max-min ant system (MMAS) for vehicle routing problem with time windows," *KKU Engineering Journal*, vol. 38, no. 3, pp. 313–323, 2011.