Transportation Research Part E 47 (2011) 166–181

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

An ant colony optimization model: The period vehicle routing problem with time windows

Bin Yu, Zhong Zhen Yang *

Transportation Management College, Dalian Maritime University, Dalian 116026, PR China

ARTICLE INFO

Article history:
Received 20 December 2009
Received in revised form 27 July 2010
Accepted 1 September 2010

Keywords:
Period vehicle routing problem with time windows
Ant colony optimization
Multi-dimension pheromone
Crossover operation

ABSTRACT

This paper proposes an improved ant colony optimization (IACO) to solve period vehicle routing problem with time windows (PVRPTW), in which the planning period is extended to several days and each customer must be served within a specified time window. Multi-dimension pheromone matrix is used to accumulate heuristic information on different days. Two-crossover operations are introduced to improve the performance of the algorithm. The effectiveness of IACO is evaluated using a set of well-known benchmarks. Some of the results are better than the best-known solutions. Results also show the IACO seems to be a powerful tool for PVRPTW.

1. Introduction

Vehicle routing problem (VRP), related to many real-life applications, holds an important place in operation research (Laporte, 2009). Vehicle routing problem with time windows (VRPTW) is a well-known generalization of VRP where each customer must be served within a specified time window. VRPTW has been widely studied both in theoretical researches and practice applications in the last 20 years (Bräysy and Gendreau, 2005a,b; Liu et al., 2009).

In VRP or VRPTW, each customer is visited exactly once during the same planning period (a single day). Differing from standard VRP or VRPTW, the VRP with multiple periods (e.g. 1 week) is defined as PVRP. PVRP has extensive real-world applications, including the collection of waste, the distribution of equipment for intermodal operations, elevator maintenance and repair, and vending machine replenishment (Francis et al., 2008).

Early heuristics for PVRP are proposed by Beltrami and Bodin (1974), Russell and Igo (1979). Chao et al. (1995) presented a two-stage heuristic which consisted of initial solution construction phase and solution improvement phase. In the first phase, they assigned delivery day combinations to customers by solving an integer programming problem. Then, some strategies were used to improve the solution. Baptista et al. (2002) presented a case study of PVRP which was about the collection of recycling paper containers in the Almada Municipality, Portugal. Compared with classical PVRP, a novel feature of their case was that the frequency of visits to containers was considered as a decision variable in their model. Similarly, Francis et al. (2006) presented a special PVRP in which service frequency of customer was variable. They modeled the PVRP with service choice (PVRPSC) in which service frequency was a decision variable, and developed an exact solution method with heuristic variations for PVRPSC. Matos and Oliverira (2004) applied the Ant System in PVRP and presented a new updating strategy of pheromone information. Then, they tested their algorithm by a waste collection system involving 202 locations in

* Corresponding author.
E-mail addresses: minlfish@yahoo.com.cn (B. Yu), yangzhongzhen@263.net (Z.Z. Yang).

1366-5545/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.tre.2010.09.010

the municipality of Viseu, Portugal. Other heuristics for PVRP were developed by Paletta (1992, 2002); Blakeley et al. (2003), Bertazzi et al. (2004), Drummond et al.(1999), Hadjiconstantinou and Baldacci (1998), Russell and Gribbin (1991).

The focus of this paper is on the period vehicle routing problem with time windows (PVRPTW), which is a generalization of PVRP. In PVRPTW, each customer can be served more than once within a specified time window during the planning period of several days. To the best of our knowledge, there is little literature to use heuristics for PVRPTW.

Ant colony optimization (Dorigo et al., 1996) is an artificial intelligence procedure inspired from food-seeking behaviors of ant colonies in nature. It has been successfully applied to some classic compounding optimization problems, e.g. traveling salesman (Dorigo et al., 1996; Stützle and Hoos, 2000), telecommunication routing (Schoonderwoerd et al., 1997), product design (Albritton and McMullen, 2007), etc. Recently, it has also been applied to solve VRP or VRPTW. Bullnheimer et al., 1997; Bullnheimer et al., 1999 presented an Ant System to solve the vehicle routing problem; Bell and McMullen (2004) presented an ACO with multiple colonies for VRP; Yu et al., 2009; Yu et al., in press presented an improved ACO, introduced ant-weight updating strategy and mutation operation, to solve VRP. Further researches were interested in ACO for VRPTW. Chen and Ting (2006) presented a hybrid algorithm combined ant colony system (ACS) and simulated annealing algorithm for VRPTW. Gong et al. (2007) presented a two-generation ACS for VRPTW. In their algorithm, the sub-tours were constructed during the child generation, while the feasible route was constructed during the parent generation. Tan et al. (2005) presented an improved ACS to solve VRPTW, in which two respective ant colonies were used to identify a multiple objective minimization. Zhu and Zhen (2009) presented a hybrid ACS with dynamic sweep algorithms for VRPTW. Qi and Sun (2008) proposed an ACS with a randomized algorithm for VRPTW.

This paper aims to test the feasibility of ACO in PVRPTW. The remainder of the paper is organized as follows. In Section 2 we describe PVRPTW. In Section 3, ACO and some improvement strategies are presented. In Section 4, some computational results are discussed and lastly, the conclusions are provided in Section 5.

2. Problem description

Standard PVRPTW can be considered to have a fleet of identical vehicles starting, serving customers and ending at the depot in each day; every customer node can be visited exactly once in the same day; customer demands can be satisfied in the period and can be serviced according to its service frequency. Moreover, service time of each customer must meet the time window constraints. An implicit constraint on the tours is that the total demand of a tour cannot exceed vehicle capacity.

PVRPTW is different from VRPTW in that the planning period for all customers is not a single day but a period of several days. The service frequency of each customer could be different, e.g., either once, twice, or three times. Fig. 1 shows an example for PVRPTW. The planning period is set to 3 day. There is a depot and 15 customers in the example. The service frequency of each customer could be once, twice, or three times, i.e., each type of customers has different service schedules.

For PVRPTW, it is important how to serve all the customers during period under the capacity and time constrains, etc. Firstly, the combination of customers on each day should be determined. Then, for the customers required service on each day the routes of vehicles are optimized like standard VRPTW, respectively. From the literature on VRPTW, there are two

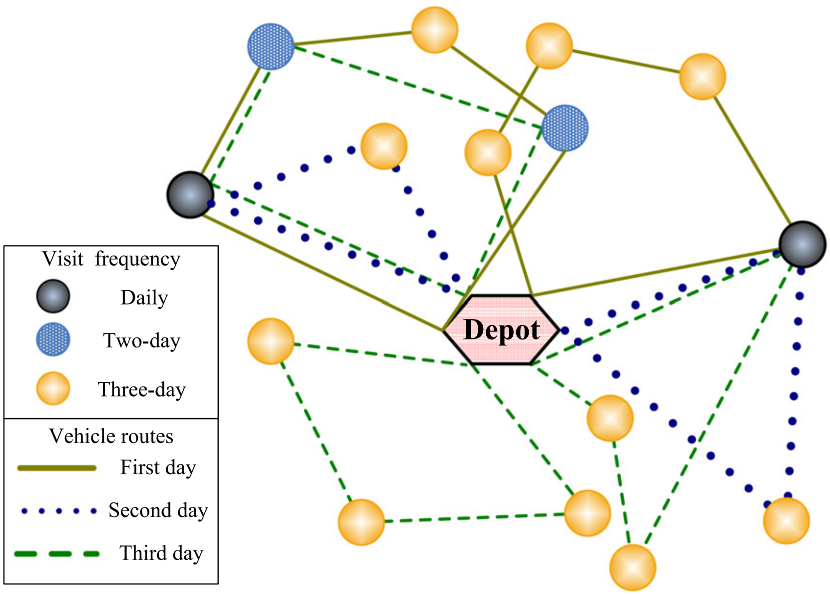

Fig. 1. An example for period vehicle routing problem.

main objectives: minimizing the number of tours first, then minimize the distance. In this study, the objective of PVRPTW is to minimize the total transportation distance of all the vehicles during period.

3. Improved ant colony optimization (IACO)

3.1. Generation of solutions

When using ACO in PVRPTW, each ant starts from the depot, constructs its route by incrementally selecting customers and returns to the depot. The customer selection is based on a probabilistic rule taking into account of both the visibility and the pheromone information. However, PVRPTW is totally different from VRP with a single day. Heuristic information (the pheromone information) of a customer during different day is greatly different, i.e., that a customer is visited during different day has a great influence on the solution. Therefore, based on the multiple days it is necessary to construct the multi-dimension pheromone information on each edge, i.e., there is different pheromone information on each edge during different days. Multiple pheromone information matrixes should be maintained during search. The number of pheromone information matrixes equals to the planning period. ACO searches routes in each day according to the pheromone information matrix of each day. Thus, an ant will move from the customer i to the customer j with probability:

$$p_{(i,j)}^{h} = \begin{cases} \dfrac{\left(\tau_{(i,j)}^{h}\right)^{\alpha} \times \left(\eta_{(i,j)}\right)^{\beta}}{\sum_{l \notin tabu} \left(\tau_{(i,l)}^{h}\right)^{\alpha} \times \left(\eta_{(i,l)}\right)^{\beta}} & j \notin tabu \\ 0 & otherwise \end{cases} \tag{1}$$

where h denotes a day; $p_{(i,j)}^{h}$ denotes the probability of choosing to combine customers i and j on the tour in the h day; $\tau_{(i,j)}^{h}$ denotes the pheromone concentration on the edge (i,j) in the h day. It can tell us how good the combination of two customers i and j was in previous iterations; $\eta_{(i,j)}$ denotes the visibility on the edge (i,j) which is defined according to the information of the problem. In this study $\eta_{(i,j)} = \frac{1}{d_{ij}}$, where, d_{ij} denotes the distance between customers i and j. α and β the weight parameters which denote the relative influence of the pheromone trails and the visibility values, respectively; *tabu* denotes the set of the customers for which no further visits are more needed in the period considered.

3.2. Crossover operation

In this paper, the crossover operations are introduced into ACO to explore search space and to prevent local optimization. For PVRPTW, the crossover operations alter two tours at the same day and renew the solution. There are two-crossover operations in our algorithm: one-point crossover operation and two-point crossover operation. The one-point crossover operation is a sub-tour exchange procedure between two tours in the same day. The two-point crossover operation is an edge exchange procedure of two tours in the same day, which search optimal crossover point combination. In two-crossover operations, all tours in the same day are tested until there is no feasible crossover that can improve the current solution. For the tours in a given day in the solution, the processes of two-crossover operations are as below.

– One-point crossover operation
  Step 0. Initialize.
  Step 1. Set the day.

If reaching the end of the planning period, goto Step 8; otherwise *day*++. *FirstCrossoverTour* = 0, *SecondCrossoverTour* = 1, *FirstCrossovePoint* = 0, *SecondCrossoverPoint* = 1.

Step 2. Select the first crossover tour.

If all the tours are tested during the current day, go to Step 1; otherwise *FirstCrossoverTour* ++.

Step 3. Select the first crossover point in the first crossover tour.

If reaching the end of the first crossover tour, go to Step 2; otherwise *FirstCrossoverPoint*++.

Step 4. Select the second crossover tour.

If all the tours are tested during the current day, go to Step 3; otherwise *SecondCrossoverTour* ++.

Step 5. Select the second crossover point.

If reaching the end of the second crossover tour go to Step4; otherwise *SecondCrossoverPoint* ++.

Step 6. Implement one-point crossover operation.

As is the one-point crossover operation in genetic algorithm, two new tours can be obtained from the crossover operation. Each selected tour is firstly broken into two segments: the beginning part which contains the crossover point and the remaining part of the tour. Then, the first new tour is obtained by combining the beginning part of the first selected tour with the ending part of the second selected tour. At the same time, the second new tour can be obtained by inversing the above operation. An example for the one-point crossover operation is as shown in Fig. 2.

We use two low tours in the first day of the example in Fig. 1 to illustrate the one-point crossover operation. For example, the points k and k' are selected as crossover points (Fig. 3), the result of the two-point crossover operation is shown as Fig. 4.

If the two new tours can satisfy the capacity constraints and time window constraints, go to Step 7; otherwise, go to Step 8.

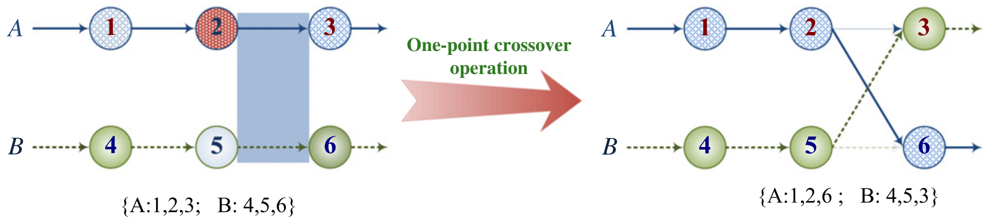

Fig. 2. An example for one-point crossover operation.

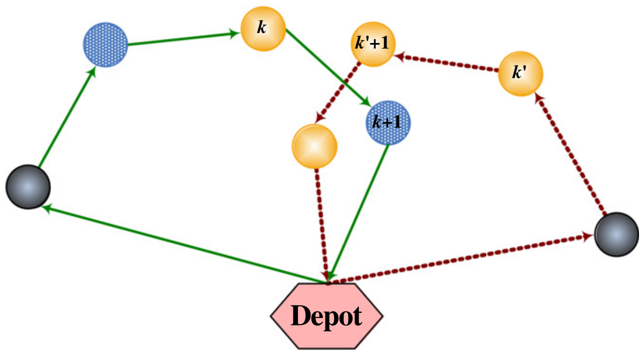

Fig. 3. Selecting the crossover points in one-point crossover operation.

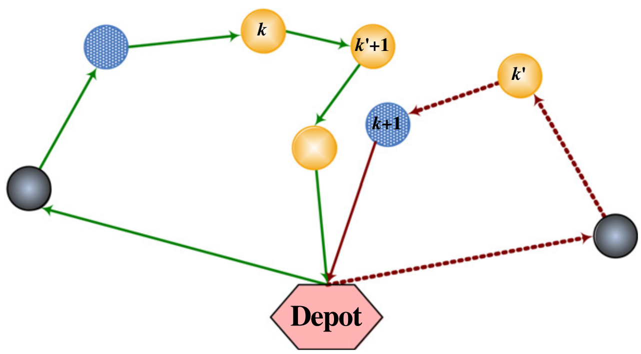

Fig. 4. Exchanging the points in one-point crossover operation.

Step 7. 2-opt operation.

For the two new tours, 2-opt algorithm (Croes, 1958) is used to ensure local optimality and improve the tours. For each tour, all possible pairwise exchanges of customer locations visited by individual vehicles are tested to see if an overall improvement in the objective function can be attained. The method has been used in several ACOs (Bullnheimer et al., 1997, 1999; Bell and McMullen, 2004; Chen and Ting, 2006) for the VRP. If the two new tours can improve the solution, the new tours replace the old tours; otherwise, cancel the one-point crossover operation.

Step 8. Termination criterion.

If there is no feasible crossover or if reaching the given iterative times, the one-point crossover operation is terminated; otherwise, go to Step 1.

– Two-point crossover operation
Step 0. Initialize.
Step 1. Set the day.
Step 2. Select the first crossover tour.
Step 3. Select the first crossover edge in the first crossover tour.
Step 4. Select the second crossover tour.
Step 5. Select the second crossover edge.
Step 6. Implement two-point crossover operation.

In the two-point crossover operation, all crossover point set, which consists of the four points of the two crossover edges, is tested to find local optimal. We select the best set, which can shorten the most length of the solution, from all feasible set of the crossover points to construct the new tours. For example, there are two tours: tour *A* and tour *B* in the same day (Fig. 5). The edge (1, 4) in the tour *A* and the edge (2, 3) in the tour *B* are the crossover edges. If there is no constraint, the four points 1, 2, 3 and 4 can construct 256 ($4 \times 4 \times 4 \times 4 = 256$) permutations. To decrease the computation time, we assume that the order of the two points in the same original tour is fixed, i.e., if the point 1 and the point 4 is in the same new tour the point 1 will be in front of the point 4 and if the point 2 and the point 3 in the same new tour the point 2 will be in front of the

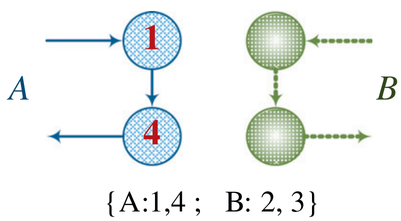

Fig. 5. Original tours before two-point crossover operation.

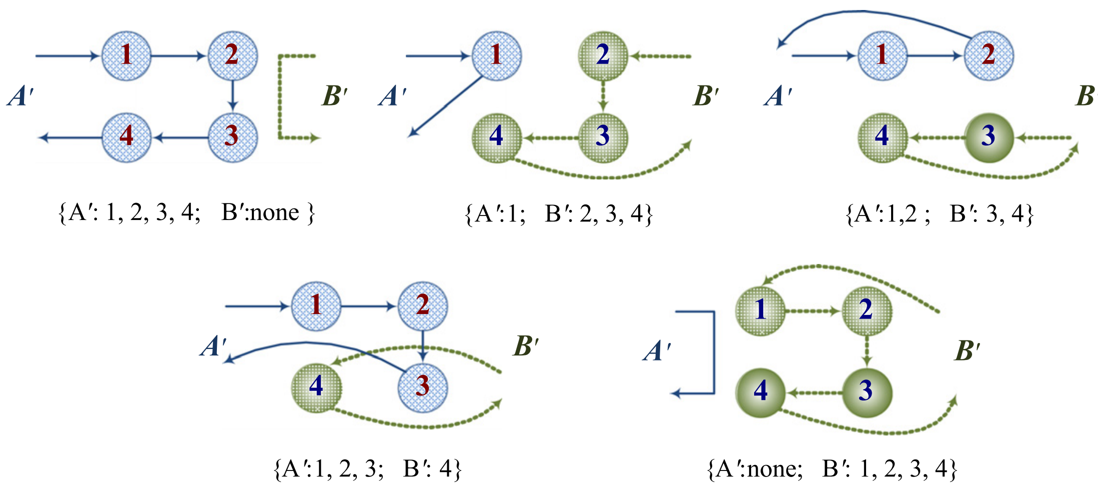

Fig. 6. Feasible combinations for the permutation {1, 2, 3, 4} in two-point crossover operation.

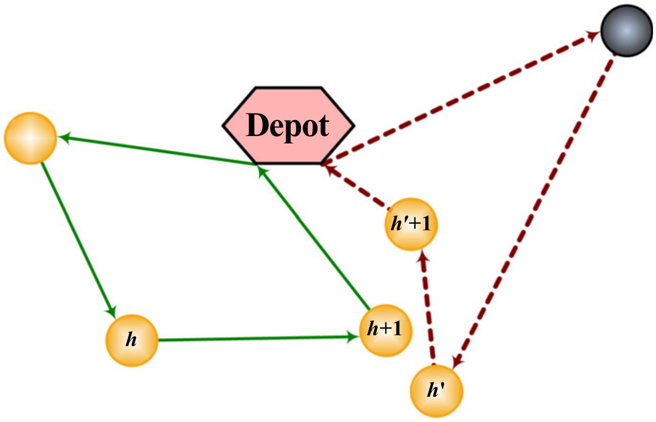

Fig. 7. Selecting the crossover edges in two-point crossover operation.

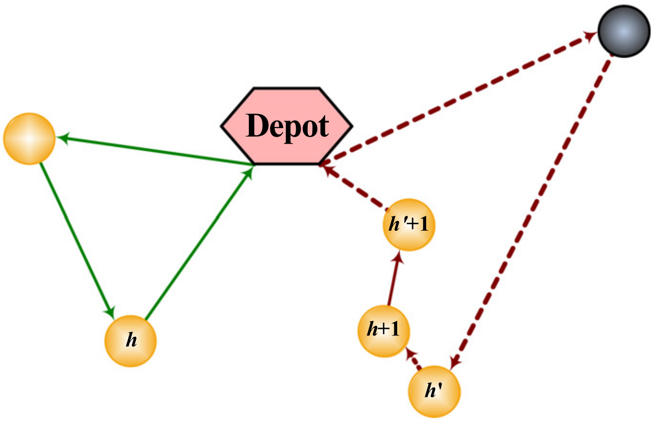

Fig. 8. Exchanging the edges in two-point crossover operation.

point 3. Thus, the number of the permutations is decreased to 6, i.e., {1, 4, 2, 3}, {1, 2, 4, 3}, {1, 2, 3, 4}, {2, 1, 4, 3}, {2, 1, 3, 4} and {2, 3, 1, 4}. After determining the permutation, we should test the various solution combinations in which the tour *A* includes a few points of the four points and the tour *B* includes the other points. For example, in the permutation {1, 2, 3, 4}, there are five feasible combinations for the new tour *A′* and the new tour *B′*: {*A′*: 1, 2, 3, 4; *B′*: none}, {*A′*:1; *B′*: 2, 3, 4}, {*A′*:1, 2; *B′*: 3, 4}, {*A′*:1, 2, 3; *B′*: 4} and {*A′*: none; *B′*: 1, 2, 3, 4} (Fig. 6).

Then, the combination that can save the most length is selected among 30 combinations (the number of the point permutation $\times$ the number of the solution combinations = $6 \times 5 = 30$) crossover solutions.

We use two low tours in the third day of the example in Fig. 1 to illustrate the two-point crossover operation. For example, the first crossover edge is $(h, h + 1)$, and the second crossover edge is $(h', h'{+}1)$ (Fig. 7). The result of the two-point crossover operation is shown as Fig. 8.

If the two new tours can satisfy the capacity constraints and time window constraints, go to Step 7; otherwise, go to Step 8.

Step 7. 2-opt operation.

The 2-opt algorithm is also used to ensure local optimality of the new tours. If the two new tours can improve the solution, the new tours replace the old tours; otherwise, cancel the two-point crossover operation.

Step 8. Termination criterion

If there is no feasible crossover or if reaching the given iterative times, the two-point crossover operation is terminated; otherwise, go to Step 1.

3.3. Update of pheromone information

The updating of the pheromone trails is a key element to the adaptive learning technique of ACO and the improvement of future solutions. To simulate the natural evaporation of pheromone information, an evaporation factor is introduced to reduce the amount of pheromone information on all edges. This is done with the following pheromone information updating equation:

$$\tau_{(i,j)}^{h\ \ new} = (1-\rho)\times\tau_{(i,j)}^{h\ \ old} + \Delta\tau_{(i,j)}^{h} \quad \rho \in (0,1) \tag{2}$$

where $\tau_{(i,j)}^{h\ \ new}$ denotes the pheromone information on the edge (i,j) after updating in the h day; $\tau_{(i,j)}^{h\ \ old}$ denotes the pheromone information on the edge (i,j) before updating in the h day; and ρ denotes the evaporation factor of the pheromone information; $\Delta\tau_{(i,j)}^{h}$ denotes the increased pheromone information on the edge (i,j) in the h day.

$$\Delta\tau_{(i,j)}^{h} = \left(\lambda \times \frac{L^{opt}}{L^{current}} \times \delta\right) \tag{3}$$

where $L^{current}$ denotes the total length of all tours in the solution during the planning period; L^{opt} denotes the total length of all tours in the optimal solution during the planning period so far; λ is a constant, which denotes the baseline increment of pheromone information; δ is the punishment coefficient, which is used to decrease the pheromone increment of the solutions where the number of the vehicles of a certain day or several days is more than the number of the given vehicles. The punishment coefficient for the infeasible solutions is introduced:

$$\delta = \frac{1}{\sum_h \varphi^h + 1} \tag{4}$$

where φ^h is the excess with respect to the maximum vehicles in the h day.

In PVRPTW, which day each customer can be served is unknown and the customer set in each day as well. This can induce some infeasible solutions that the fleet constraint cannot be satisfied in one or a few days during planning period.

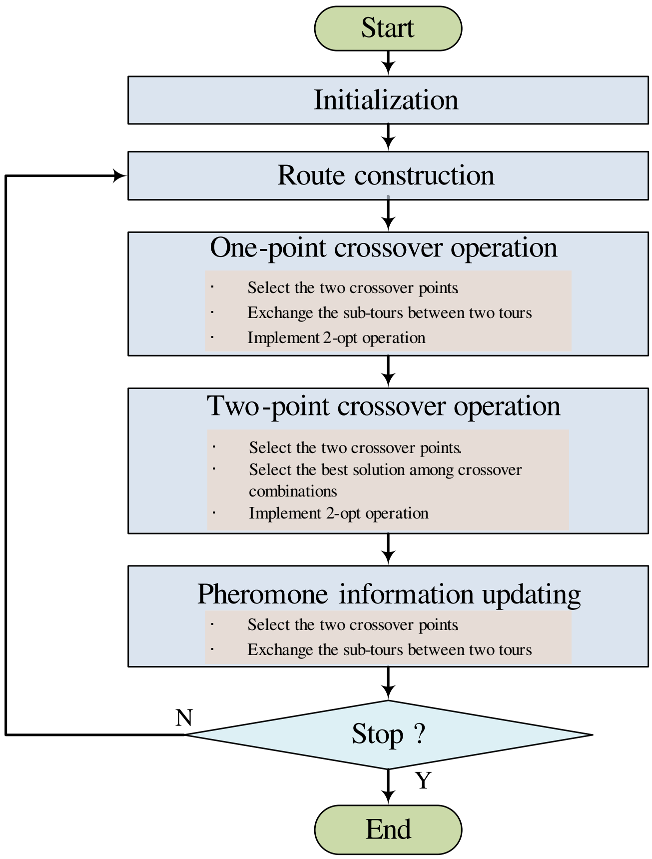

Fig. 9. The flowchart of the searching process for IACO.

$$\varphi^h = \begin{cases} J^h - K & \text{if } J^h > K \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where J^h represents the number of vehicles in the h day in the solutions.

In addition, to avoid sub-optimization and to enlarge the probability of obtaining higher-quality solution, upper and lower limits $[\tau_{min}, \tau_{max}]$ are fixed to the updating equation. The initial pheromone information of each edge is set to the maximum pheromone information. In this study, the upper and lower limits are set as follow:

$$\tau_{\min} = 1 \tag{6}$$

$$\tau_{\max} = N \tag{7}$$

where N denotes the maximum iteration times of the improved ACO.

In ACO search, the evaporation factor (ρ) and the baseline pheromone increment coefficient (λ) determine the pheromone distribution in search space. The two parameters for various problems are different and difficult to be determined. However, it can be fixed that the evaporation pheromone information of each iteration is $\rho \times \tau_{(i,j)}^{h\ \ old}$. Thus, to ensure the attraction of new solutions, the pheromone increment should be at least comparable to the evaporation pheromone information of each iteration. In this paper, the baseline pheromone increment coefficient is set to $\lambda = \tau_{\max} \times \rho$.

3.4. Overall procedure

The flowchart of our IACO for the PVRPTW is shown in Fig. 9.

4. Numerical analysis

Since we cannot find the literature on heuristics for PVRPTW so far, it is difficult to compare our algorithm with previous methods. In this section, sensitivity analyses of parameters in IACO are performed and the performances of several improvement strategies are also tested.

We tested our algorithm on some benchmark instances. These instances and the best-known solutions are available at http://neo.lcc.uma.es/radi-aeb/WebVRP/. The information on these instances is shown in Table 1.

The main characteristics of these benchmark instances are summarized in Table 1.

The heuristics described in the previous sections were coded in Visual C++.Net 2003 and executed on a PC equipped with 3.25 GB of RAM and a Pentium processor running at 2.93 GHz.

4.1. Sensitivity analyses of parameters

The parameter selection is important for the performance of the algorithm. In this section, the parameters in our algorithm are firstly tuned, in which there are three parameters: the control factor of the pheromone trails α, the control factor of the visibility values β and the evaporation factor ρ. Referring to previous literature using ACO in VRP, the ranges of three parameters were set to $\alpha \in \{1, 2, 3\}$, $\beta \in \{2, 4, 6\}$, $\rho \in \{0.1, 0.05, 0.01\}$. When tuning the parameters, the instance Pr01 was determined as the test problem. Then, the algorithm with each parameter combination for the instance Pr01 was tested 10 times.

From the test results, it can be found that the algorithms, in which evaporation factor is set to 0.01, can yield better solutions. This can be attributed to that the smaller evaporation factor can ensure the sufficient diversity of search space and guide following ants to explore better solutions. Otherwise, if pheromone evaporation is too rapid, it is more easily to result in the search to be trapped in local minima. Pheromone trail represents heuristic information of previous search and visibility values represent desirability represents fixed information. Based on the results of the tests, the algorithm with the smaller weight parameter (α) of pheromone trails possesses higher performance. Furthermore, with the increase of the weight parameter of pheromone trails ($\alpha > 3$), the performance of the algorithm greatly decreases. This

Table 1
Characteristics of the 10 benchmark problems.

No.	n	Q	Period (day)	The number of vehicles	No.	n	Q	Period (day)	The number of vehicles
Pr01	48	200	4	3	Pr06	288	175	4	18
Pr02	96	195	4	6	Pr07	72	200	6	5
Pr03	144	190	4	9	Pr08	144	190	6	10
Pr04	192	185	4	12	Pr09	216	180	6	15
Pr05	240	185	4	15	Pr10	288	170	6	20

Table 2
Best computational results for PVRPTW with IACO on 10 test problems.

No.	Best-known solution		Best solution of IACO			
	The number of tours in each day	Length	The number of tours in each day	Length	Average time	Deviation (%)
Pr01	3 + 3 + 3 + 3 = 12	3007.84	3 + 3 + 3 + 3 = 12	2959.09	171.56	−1.62
Pr02	6 + 5 + 5 + 6 = 22	5328.33	6 + 6 + 6 + 6 = 24	5323.29	339.57	−0.09
Pr03	9 + 9 + 9 + 9 = 36	7397.10	9 + 9 + 9 + 9 = 36	7554.5	609.79	2.13
Pr04	10 + 11 + 10 + 9 = 40	8376.95	12 + 10 + 8 + 9 = 39	8364.61	1742.82	−0.15
Pr05	13 + 13 + 13 + 11 = 50	8967.90	12 + 15 + 10 + 13 = 50	8964.46	1813.50	−0.04
Pr06	15 + 14 + 15 + 18 = 62	11686.91	18 + 14 + 12 + 12 = 56	11122.6	2872.44	−4.83
Pr07	5 + 5 + 5 + 5 + 5 + 5 = 30	6991.54	5 + 5 + 5 + 5 + 5 + 5 = 30	7100.24	530.10	1.55
Pr08	8 + 9 + 8 + 8 + 8 + 8 = 49	10045.05	8 + 10 + 8 + 6 + 8 + 6 = 46	10094.58	849.06	0.49
Pr09	13 + 11 + 12 + 13 + 11 + 11 = 71	14294.97	13 + 13 + 12 + 12 + 10 + 9 = 69	14356.9	2676.06	0.43
Pr10	16 + 15 + 16 + 17 + 16 + 16 = 96	18609.72	12 + 16 + 15 + 15 + 13 + 20 = 91	17733.2	3802.68	−4.71
Average	46.8	9470.631	41.3	9357.347	1542.56	−1.20

may be attributed to that in our algorithm the initial pheromone trails are large values. If using the large control factor of pheromone trail, the effect of visibility value is weakened and results in a premature convergence. In addition, the qualities of the solutions of the algorithms with $\beta = 3$ and $\beta = 6$ are similar. However, considering the computation time the control factor of visibility value is set to 6. Thus, the best parameter combination is determined: $\{\rho = 0.01, \alpha = 2, \beta = 6\}$.

4.2. Benchmark Instances

After the parameter determination, our algorithm is tested for the other instances. The results of our algorithm for all the instances are shown in Table 2. It can be observed that our algorithm can obtain better solutions than the best-known solutions for most of the instances. Our algorithm achieves six better solutions among 10 instances than best-known solutions. However, for the instances Pr03, and Pr07, our algorithm cannot obtain the best-known solutions. This can be due to that there are tighter constraints for vehicle fleet in the instance Pr03 and Pr07. For the instances with tight fleet constraint, our algorithm often finds infeasible solutions during search, i.e., the number of vehicles in a certain day of the solution is more than the given vehicles during period. Thus, our algorithm cannot adequately search the solution space and the best solutions for the instances of our algorithm are worse than the best-known solutions. However, for the instances with the loose fleet constraint, e.g., the instances Pr06, Pr08, Pr09 and Pr10, our algorithm can achieve the better or comparable solutions, especially, for the instances Pr06 and Pr10. In addition, it can be found that the running time greatly increase along with the increase of the number of customers, e.g. the running time of the instance Pr10 is about 3802.68s.

From Table 2, it can also be observed that our algorithm seems to be superior in terms of solution quality with an average deviation of -1.20%. This induces that our algorithm, which includes the multi-dimension pheromone information, the one-point and two-point crossover operations, has an improved performance for PVRPTW. To test the improvement strategies, we also design two cases.

– Multi-dimension pheromone information

To validate the efficiency of the multi-dimension pheromone information, two ACO are constructed: ACO with the multi-dimension pheromone information (denoted by ACO+MP) and ACO with the single-dimension pheromone information (denoted by ACO+SP). Results show that ACO with the single-dimension pheromone information (denoted by ACO+SP) is unsuitable for solving the PVRPTW. This can be due to that ACO+SP uses the single-dimension pheromone information regardless of the periodic service features, which makes difficult to satisfy the fleet constraint. On the other hand in ACO+MP, the multi-dimension pheromone information can maintain and update heuristic information of each customer in different days. Although the solutions from ACO+MP are greatly worse than the best-known solutions, it can still search the feasible solutions for all instances. This indicates that the multi-dimension pheromone information is an efficient strategy to reduce infeasible solutions during search.

– Crossover operation

To evaluate the crossover operations, three ant colony optimizations with different strategies are also constructed. The first one is ACO+MP (with no local search), the second one is an ACO+MP with the one-point crossover operation (denoted by ACO-MPO) and the last one is an ACO+MP with the two-point crossover operation (denoted by ACO-MPT). The results are compared with those obtained with IACO, that has both crossover operations. For each ACO, the search will be stopped as the solution cannot be improved 100 consecutive generations or the search reaches the maximization generation (10,000 gen-

Table 3
Comparisons of ACO with several crossover operations.

Problem		Pr01	Pr02	Pr03	Pr04	Pr05	Pr06	Pr07	Pr08	Pr09	Pr10
ACO+MP	Best solution	3518.36	6395.25	10981.47	10081.38	10535.84	12958.24	9674.18	12625.48	19597.2	21102.91
	Average solution	3905.38	7226.63	12079.62	11492.77	11694.78	14901.98	11028.57	14898.07	22732.75	24690.40
	Average time	130.85	281.69	504.26	1443.89	1443.44	2393.06	421.61	708.38	2064.99	3299.99
ACO-MPO	Best solution	3189.08	5710.3	8630.81	9217.8	9758.22	12011.32	8453.11	11719.92	16882.85	19730.55
	Average solution	3377.24	6007.24	9407.58	10323.94	10831.62	13813.02	9687.26	13595.11	18739.96	22098.22
	Average time	150.47	301.17	544.7	1527.74	1553.62	2590.59	470.31	764.35	2312.18	3489.93
ACO-MPT	Best solution	3053.78	5593.64	8480.91	8715.92	9376.83	11656.48	7917.57	11219.97	16002.67	18982.53
	Average solution	3215.63	5873.32	9413.81	9674.67	10220.74	12938.69	8867.68	12858.09	17922.99	21450.26
	Average time	162.29	329.25	590.36	1672.96	1748.51	2772.52	517.7	819.6	2534.44	3701.13
IACO	Best solution	2959.09	5323.29	7554.5	8364.61	8964.46	11122.6	7100.24	10094.58	14356.9	17733.2
	Average solution	3107.04	5658.66	8158.86	9117.42	9591.97	12346.09	8023.27	11305.93	15936.16	19151.86
	Average time	171.56	339.57	609.79	1742.82	1813.5	2872.44	530.1	849.06	2676.06	3802.68

erations). Then, the ACOs continue running ten times under the same condition. The best solutions, average solution and average computation time are shown as Table 3 over these 10 runs.

It is obvious that the running time of ACO+MP is the least of all, while the solution quality of ACO+MP is the worst among four algorithms. This can be attributed that ACO+MP without any improvement strategies cannot adequately search the solution space and obtain better solutions. Actually the local search operations seem to behave quite nicely, the computational times are not so greatly increased. Furthermore, this is just as expected that ACO+MPT can generally provide better solutions than ACO+MPO. It is can be explained that two-point crossover operation implements more operations and enlarge the search space. However, since these operations in the two-point crossover operation need more time to implement, it can also be observed that the running time of ACO+MPT is less than the one of ACO+MPO. In addition, compared with the average solution of several algorithms, the stabilization performance of IACO, in which the difference between the best solution and the average solution is about 9%, is the best. The stabilization performances of ACO+MPT and ACO+MPO are similar, and they are better than the one of ACO+MP. Overall, the performance of our algorithm with the multi-dimension pheromone information, one-point and two-point crossover operations is the best among four algorithms. This indicates that the incorporation of two-crossover operations can improve the performance of ACO.

5. Conclusions

Period vehicle routing problem with time windows (PVRPTW) is a generalization of classical vehicle routing problem, in which the planning period is extended from a single day to several days and each customer must be served within a specified time window. This paper proposes an improved ant colony optimization (IACO) with multi-dimension pheromone information and two-crossover operations to solve PVRPTW. The effectiveness of the improved ant colony optimization is evaluated using a set of well-known benchmarks. Computational results show that the incorporation of multi-dimension pheromone information and two-crossover operations can improve the performance of ACO for PVRPTW.

The main contributions of this paper to the literature can be summarized as follow:

1. This paper tests the feasibility of ACO for PVRPTW.
2. When applying IACO to solve PVRPTW, multi-dimensional pheromone information is the most suitable extension, which can be used to distinguish heuristic information during different days from previous searches.
3. Furthermore, the two-crossover operation always outperforms the one-crossover operation in the search quality, while the two-crossover operation need more computation time.
4. The PVRPTW is a problem not much explored, but it has many possible applications in real world problems. This paper aims to formulate an approach and solution to stimulate the attention of other researchers.

Acknowledgments

This research is financed by the National Science Foundation for Post-doctoral Scientists of China 20080440168, the Doctoral Program Foundation for Young Scholar of Institutions of Higher Education of China through project 20070151013 and the Special Fund for Basic Scientific Research of Central Colleges, Dalian Maritime University 2009QN094. The authors would also like to thank Dr. SUN Jian from School of Transportation Engineering, Tongji University for supporting his efforts for this paper.

Appendix A

Problem	Length	Day number	Tour number	Tour
Pr01	2959.09	1	1	0-47-24-12-15-5-29-20-33-8-1-28-4-0
		1	2	0-41-36-2-32-31-44-0
		1	3	0-9-10-11-6-3-45-22-7-35-0
		2	1	0-9-12-21-30-2-23-26-18-17-16-0
		2	2	0-10-11-6-3-34-42-46-7-0
		2	3	0-14-5-8-13-1-4-0
		3	1	0-5-20-1-19-4-8-0
		3	2	0-15-24-12-38-40-25-2-39-43-0
		3	3	0-9-37-10-48-11-27-3-6-22-7-0
		4	1	0-9-12-21-2-18-17-23-0
		4	2	0-10-6-3-11-7-0
		4	3	0-14-5-8-13-1-16-4-0
Pr02	5323.29	1	1	0-22-42-85-56-12-3-83-40-4-77-0
		1	2	0-72-60-73-37-9-47-0
		1	3	0-18-14-80-24-6-10-2-1-0
		1	4	0-95-89-52-61-91-46-21-54-11-0
		1	5	0-15-23-90-13-8-70-0
		1	6	0-65-17-64-7-5-16-20-19-81-0
		2	1	0-15-30-23-34-49-75-21-11-0
		2	2	0-32-94-17-44-9-1-0
		2	3	0-22-71-12-3-36-10-2-33-0
		2	4	0-29-28-26-45-35-31-4-59-25-0
		2	5	0-58-41-43-13-38-27-68-8-19-0
		2	6	0-18-14-24-6-48-7-5-39-16-20-0
		3	1	0-57-79-63-13-8-0
		3	2	0-22-66-47-1-0
		3	3	0-67-14-18-24-6-50-12-3-76-51-2-78-88-0
		3	4	0-15-23-46-82-4-40-21-11-0
		3	5	0-42-96-10-7-5-16-20-74-19-0
		3	6	0-17-62-84-86-93-92-55-37-69-9-87-0
		4	1	0-15-30-23-34-35-26-45-4-31-21-11-0
		4	2	0-29-0
		4	3	0-32-17-44-38-27-39-20-19-0
		4	4	0-22-12-3-53-36-10-2-33-1-0
		4	5	0-28-41-43-13-8-25-0
		4	6	0-18-14-24-6-48-7-5-16-9-0
Pr04	8364.61	1	1	0-115-99-163-164-44-185-17-37-100-78-184-87-119-92-144-0
		1	2	0-25-110-150-35-2-30-46-60-95-21-5-0
		1	3	0-113-69-36-98-22-0
		1	4	0-90-84-136-6-29-54-156-102-43-26-96-61-15-154-0
		1	5	0-187-40-20-105-1-86-120-158-140-168-0
		1	6	0-27-10-47-39-167-130-16-34-53-109-132-52-24-134-0
		1	7	0-135-190-38-123-188-14-186-0
		1	8	0-91-157-3-137-11-59-19-64-93-0
		1	9	0-171-31-4-23-45-146-179-48-75-126-121-68-0
		1	10	0-148-12-174-74-7-80-176-155-13-72-70-0
		1	11	0-153-28-63-49-124-41-94-8-143-33-0
		1	12	0-180-50-71-32-42-56-177-9-82-161-18-0
		2	1	0-25-108-57-122-73-149-81-7-46-183-116-142-14-0
		2	2	0-12-89-131-13-138-106-66-0
		2	3	0-145-129-97-47-39-16-34-62-77-36-22-0
		2	4	0-165-32-9-42-118-141-67-104-0
		2	5	0-147-40-20-4-23-45-170-1-101-33-0
		2	6	0-38-3-79-178-107-111-11-125-58-128-133-173-0
		2	7	0-28-41-103-114-6-29-169-175-8-83-189-18-0

Appendix A (*continued*)

Problem	Length	Day number	Tour number	Tour
		2	8	0-27-10-162-127-55-159-85-160-24-19-139-0
		2	9	0-182-31-51-88-151-112-44-17-37-26-43-48-191-15-0
		2	10	0-192-76-181-35-2-30-21-5-117-65-152-0
		3	1	0-27-10-34-39-47-53-52-24-64-93-0
		3	2	0-91-38-3-25-7-72-74-13-14-70-0
		3	3	0-71-50-90-84-29-6-54-61-75-48-78-87-92-0
		3	4	0-69-22-36-16-59-11-19-0
		3	5	0-12-32-42-56-9-82-18-0
		3	6	0-63-49-28-41-94-8-44-23-45-17-37-43-26-96-15-68-0
		3	7	0-80-35-2-30-46-172-21-60-95-5-0
		3	8	0-31-40-20-4-86-1-33-0
		4	1	0-27-10-47-55-11-58-14-0
		4	2	0-34-16-39-85-19-24-0
		4	3	0-7-81-89-9-42-67-15-0
		4	4	0-38-3-79-73-25-57-46-30-2-13-66-0
		4	5	0-12-32-76-35-21-5-65-0
		4	6	0-31-40-20-4-1-36-0
		4	7	0-28-41-8-29-6-166-48-43-26-33-0
		4	8	0-88-51-23-45-44-17-37-83-18-0
		4	9	0-62-77-22-0
Pr05	8964.46	1	1	0-153-221-204-240-34-43-72-85-0
		1	2	0-24-38-199-45-41-31-42-0
		1	3	0-207-162-106-202-77-62-118-175-3-170-37-101-0
		1	4	0-81-211-158-157-142-217-151-0
		1	5	0-56-115-125-102-183-188-100-0
		1	6	0-29-149-44-104-48-114-57-21-33-60-28-0
		1	7	0-25-169-120-53-10-205-13-5-121-212-0
		1	8	0-213-83-116-8-226-6-206-9-12-32-0
		1	9	0-137-174-134-147-2-47-27-26-50-23-238-18-36-0
		1	10	0-80-177-161-84-11-193-203-55-20-187-35-0
		1	11	0-141-54-17-176-58-70-46-52-40-232-7-14-49-30-0
		1	12	0-59-231-4-51-39-22-16-1-19-194-129-15-0
		2	1	0-56-2-47-76-110-98-95-68-215-7-234-97-30-0
		2	2	0-105-132-18-165-0
		2	3	0-59-4-90-94-39-210-173-65-67-36-79-0
		2	4	0-54-119-71-164-46-52-40-109-155-92-178-218-0
		2	5	0-29-11-91-75-160-21-57-184-48-96-224-60-124-0
		2	6	0-24-74-6-53-31-186-167-191-42-236-0
		2	7	0-25-69-128-208-14-34-237-43-239-123-201-0
		2	8	0-166-5-108-146-136-113-87-49-182-0
		2	9	0-3-10-111-107-13-171-139-64-225-0
		2	10	0-44-8-195-45-38-89-41-78-93-82-0
		2	11	0-135-37-190-73-66-9-32-12-112-222-103-0
		2	12	0-35-235-20-230-28-197-55-0
		2	13	0-154-51-180-200-16-117-209-228-61-214-148-227-0
		2	14	0-50-88-58-159-86-220-22-223-19-1-33-131-15-0
		2	15	0-99-17-63-127-27-26-196-23-163-144-143-0
		3	1	0-77-25-138-53-120-152-126-42-13-140-30-0
		3	2	0-59-115-4-51-39-22-16-1-114-19-15-133-0
		3	3	0-106-62-118-3-10-12-32-5-0
		3	4	0-54-18-43-179-122-34-72-0
		3	5	0-80-8-45-38-41-31-192-9-0
		3	6	0-29-81-84-44-168-104-48-21-57-33-60-0
		3	7	0-83-116-24-185-6-172-181-28-37-0
		3	8	0-50-58-70-46-52-216-40-233-23-36-0

(*continued on next page*)

Appendix A (*continued*)

Problem	Length	Day number	Tour number	Tour
		3	9	0-56-17-26-229-27-47-2-130-7-14-49-85-0
		3	10	0-100-101-102-11-55-20-35-0
		4	1	0-29-20-35-0
		4	2	0-4-90-39-51-16-117-1-19-15-0
		4	3	0-43-99-27-2-68-14-34-0
		4	4	0-24-74-8-45-41-78-93-82-0
		4	5	0-25-69-76-47-110-98-95-7-97-49-30-0
		4	6	0-5-108-113-87-198-64-0
		4	7	0-119-71-50-88-52-46-109-40-67-92-65-0
		4	8	0-44-48-57-21-96-60-150-28-0
		4	9	0-59-61-105-0
		4	10	0-56-26-17-63-23-36-18-79-0
		4	11	0-145-3-10-111-13-107-42-103-112-9-0
		4	12	0-54-58-86-22-94-11-91-75-33-189-55-0
		4	13	0-156-53-89-38-219-31-6-73-66-12-32-37-0
Pr06	11122.6	1	1	0-229-138-88-46-162-47-45-22-161-164-6-214-1-37-225-133-0
		1	2	0-277-39-86-265-69-220-185-160-11-0
		1	3	0-99-279-175-270-130-210-95-188-127-0
		1	4	0-125-190-209-40-18-198-114-66-109-0
		1	5	0-176-118-180-264-200-8-205-68-42-34-287-0
		1	6	0-267-195-81-84-76-116-14-91-20-28-0
		1	7	0-137-74-106-167-139-228-19-55-0
		1	8	0-207-244-134-174-63-208-250-13-113-151-223-232-0
		1	9	0-56-24-10-51-165-187-75-280-61-239-227-82-0
		1	10	0-77-25-44-258-43-70-158-142-48-263-224-0
		1	11	0-90-169-213-248-59-104-147-140-41-256-194-72-60-0
		1	12	0-268-135-17-54-27-71-257-193-35-5-15-144-288-148-0
		1	13	0-159-80-149-278-128-30-33-12-21-108-129-252-0
		1	14	0-120-152-254-231-242-3-284-53-4-31-115-32-233-262-247-0
		1	15	0-145-168-83-274-2-126-237-65-96-57-23-64-181-0
		1	16	0-58-62-50-206-7-9-178-234-236-0
		1	17	0-249-16-29-219-26-52-241-102-98-246-196-36-107-191-0
		1	18	0-153-38-245-221-146-282-100-217-173-184-49-67-0
		2	1	0-251-25-276-41-112-131-72-60-0
		2	2	0-117-111-179-177-66-67-11-285-172-253-0
		2	3	0-2-50-63-73-13-238-166-33-182-215-19-55-0
		2	4	0-101-58-40-18-0
		2	5	0-156-38-17-204-202-27-35-5-235-92-222-85-171-15-272-0
		2	6	0-39-69-8-105-20-216-78-87-261-273-28-0
		2	7	0-54-71-89-240-157-31-32-79-283-269-255-49-0
		2	8	0-3-53-260-4-70-43-48-186-218-281-0
		2	9	0-56-24-10-51-61-203-57-23-0
		2	10	0-199-141-154-45-22-6-132-124-121-30-1-21-201-0
		2	11	0-16-29-36-123-259-37-192-93-189-9-97-0
		2	12	0-26-46-52-47-163-12-122-230-143-14-34-197-0
		2	13	0-44-266-59-119-275-110-136-68-42-286-103-0
		2	14	0-211-62-170-183-155-94-7-64-65-0
		3	1	0-83-134-63-50-62-126-2-113-13-36-133-0
		3	2	0-127-99-95-130-55-0
		3	3	0-16-29-26-52-46-102-98-139-33-128-30-12-108-129-19-0
		3	4	0-25-44-120-115-70-43-142-48-67-0
		3	5	0-74-84-76-14-91-243-20-28-0
		3	6	0-137-116-80-34-212-42-68-150-0
		3	7	0-135-24-51-10-56-226-61-82-57-0
		3	8	0-138-88-106-81-47-45-6-22-21-1-37-9-7-96-0

Appendix A (*continued*)

Problem	Length	Day number	Tour number	Tour
		3	9	0-125-58-18-40-23-64-65-107-0
		3	10	0-77-8-90-69-104-59-140-41-72-60-0
		3	11	0-118-38-17-54-53-4-27-71-31-32-49-109-0
		3	12	0-39-86-11-3-100-66-114-75-35-5-15-144-0
		4	1	0-38-17-54-3-53-4-89-71-27-85-35-5-15-0
		4	2	0-111-117-67-11-0
		4	3	0-46-141-93-37-9-97-0
		4	4	0-16-29-36-123-1-21-12-19-55-0
		4	5	0-87-78-14-122-143-34-28-0
		4	6	0-2-73-13-0
		4	7	0-44-70-43-48-31-32-79-92-49-0
		4	8	0-56-24-10-51-66-101-18-40-23-0
		4	9	0-58-62-50-63-7-94-61-57-64-65-0
		4	10	0-26-52-47-45-22-132-6-271-124-121-30-33-0
		4	11	0-39-69-8-20-105-41-112-131-72-60-0
		4	12	0-25-59-119-136-110-68-42-103-0
Pr10	17733.2	1	1	0-13-23-154-36-26-52-83-51-0
		1	2	0-180-61-5-3-231-31-8-0
		1	3	0-48-19-116-60-4-165-37-140-0
		1	4	0-35-249-205-28-134-1-21-43-29-262-69-0
		1	5	0-275-42-56-68-44-27-72-2-46-49-0
		1	6	0-246-64-50-58-33-70-279-14-54-0
		1	7	0-55-57-11-39-15-53-256-67-0
		1	8	0-25-257-66-115-12-45-63-41-110-128-152-0
		1	9	0-32-92-251-192-22-18-269-0
		1	10	0-126-86-0
		1	11	0-47-30-24-95-65-9-10-265-62-40-244-122-274-129-17-157-118-0
		1	12	0-20-34-16-286-7-59-71-38-6-0
		2	1	0-259-153-124-25-114-264-226-232-46-131-119-49-69-0
		2	2	0-108-32-143-22-117-85-4-37-18-0
		2	3	0-13-55-142-130-39-11-15-241-211-199-94-35-0
		2	4	0-263-103-138-19-135-113-70-123-54-14-62-0
		2	5	0-268-206-50-64-30-24-104-96-36-23-161-0
		2	6	0-20-34-151-2-42-43-141-28-0
		2	7	0-87-139-106-91-215-7-242-71-59-89-212-0
		2	8	0-267-97-218-214-216-8-31-125-194-0
		2	9	0-98-75-93-10-112-179-100-276-0
		2	10	0-181-61-159-12-73-133-27-38-72-6-21-1-145-29-0
		2	11	0-57-272-58-107-82-33-67-53-197-0
		2	12	0-109-185-79-203-230-132-102-74-105-81-250-101-0
		2	13	0-84-48-266-77-65-9-171-60-144-173-193-177-17-0
		2	14	0-5-120-198-3-136-273-137-45-283-127-221-66-0
		2	15	0-16-56-68-44-88-121-111-78-41-99-63-90-76-0
		2	16	0-47-80-52-164-26-40-51-0
		3	1	0-202-271-115-158-243-45-12-172-66-0
		3	2	0-42-2-16-222-155-27-38-72-6-208-21-43-0
		3	3	0-191-196-150-56-68-200-174-285-147-23-86-0
		3	4	0-35-28-34-134-1-258-46-49-0
		3	5	0-13-55-236-57-58-210-195-33-67-53-217-0
		3	6	0-61-186-5-167-3-270-227-183-252-31-8-0
		3	7	0-32-92-201-166-22-277-129-17-18-37-140-0
		3	8	0-261-190-188-19-182-209-238-189-70-280-235-175-14-62-0
		3	9	0-50-64-187-30-24-95-47-52-146-288-83-40-0
		3	10	0-65-48-9-260-207-10-233-160-118-122-126-0

(*continued on next page*)

Appendix A (*continued*)

Problem	Length	Day number	Tour number	Tour
		3	11	0-287-176-149-223-4-60-116-184-54-51-0
		3	12	0-25-163-168-213-0
		3	13	0-20-44-169-148-237-41-63-110-128-0
		3	14	0-240-39-11-204-162-224-245-15-239-254-36-26-0
		3	15	0-170-178-7-71-59-156-29-69-0
		4	1	0-35-0
		4	2	0-108-10-9-75-65-77-112-100-17-85-37-0
		4	3	0-180-119-0
		4	4	0-13-130-39-11-15-255-74-81-105-89-59-0
		4	5	0-109-79-44-111-132-102-99-63-90
		4	6	0-0-84-48-19-103-138-135-113-70-123-54-14-62-0
		4	7	0-87-94-34-42-43-141-28-0
		4	8	0-47-80-98-40-51-0
		4	9	0-57-107-58-82-33-67-53-0
		4	10	0-61-5-120-3-136-133-12-73-45-137-125-31-8-0
		4	11	0-124-114-25-97-127-66-46-131-205-49-0
		4	12	0-20-2-248-16-21-1-6-72-71-101-29-69-0
		4	13	0-32-117-143-192-22-93-144-60-4-165-18-157-0
		4	14	0-106-139-142-91-56-68-7-88-27-78-41-121-38-152-76-0
		4	15	0-55-24-30-64-50-104-96-154-36-26-52-23-0
		5	1	0-153-25-181-66-159-12-45-214-31-216-8-0
		5	2	0-206-50-64-58-33-70-14-54-51-0
		5	3	0-278-19-116-60-4-37-140-0
		5	4	0-61-5-3-198-194-63-41-110-38-128-0
		5	5	0-13-23-164-52-83-40-62-86-0
		5	6	0-47-30-95-24-197-67-53-26-0
		5	7	0-32-92-22-18-129-126-0
		5	8	0-185-115-203-27-16-72-6-46-49-0
		5	9	0-35-0
		5	10	0-55-57-11-39-15-211-199-212-36-161-0-
		5	11	0-65-48-9-10-171-173-193-177-17-179-118-122-0
		5	12	0-20-34-151-2-42-43-134-28-0
		5	13	0-44-68-56-7-215-59-71-21-1-145-29-69-0
		6	1	0-24-30-113-195-82-33-70-54-69-0
		6	2	0-87-142-147-94-0
		6	3	0-32-117-149-176-168-114-127-97-119-0
		6	4	0-47-55-104-234-96-52-146-35-0
		6	5	0-61-133-253-158-12-73-45-137-172-66-0
		6	6	0-68-56-150-7-121-111-78-41-99-63-90-76-0
		6	7	0-219-25-163-247-228-3-136-31-8-225-0
		6	8	0-98-10-93-9-112-160-100-0
		6	9	0-201-190-143-166-22-124-284-213-0
		6	10	0-20-28-141-43-34-1-46-131-49-0
		6	11	0-75-103-138-123-65-189-281-77-184-175-14-0
		6	12	0-13-57-130-39-11-204-162-220-15-156-23-0
		6	13	0-42-2-16-155-88-27-38-72-6-208-21-29-0
		6	14	0-64-50-229-135-80-62-40-51-0
		6	15	0-79-44-169-148-132-102-282-174-89-105-101-59-0
		6	16	0-139-106-170-178-91-191-196-200-74-81-71-0
		6	17	0-109-202-186-5-120-167-125-183-0
		6	18	0-108-84-48-19-209-182-60-144-207-17-0
		6	19	0-188-4-85-37-18-0
		6	20	0-187-58-107-210-53-67-36-26-0

References

Albritton, M.D., McMullen, P.R., 2007. Designing optimal products using a colony of virtual ants. European Journal of Operational Research 176 (1), 498–520.

Baptista, S., Oliveira, R.C., Zuquete, E., 2002. A period vehicle routing case study. European Journal of Operational Research 139 (2), 220–229.

Bell, J.E., McMullen, P.R., 2004. Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics 18 (1), 41–48.

Beltrami, E., Bodin, L., 1974. Networks and vehicle routing for municipal waste collection. Networks 4 (1), 65–94.

Bertazzi, L., Paletta, G., Speranza, M.G., 2004. An improved heuristic for the period traveling salesman problem. Computers and Operations Research 31 (8), 1215–1222.

Blakeley, F., Argüello, B., Cao, B., Hall, W., Knolmajer, J., 2003. Optimizing periodic maintenance operations for Schindler elevator corporation. Interfaces 33 (1), 67–79.

Bräysy, O., Gendreau, M., 2005a. Vehicle routing problem with time windows, part i: route construction and local search algorithms. Transportation Science 39 (1), 104–118.

Bräysy, O., Gendreau, M., 2005b. Vehicle routing problem with time windows, part ii: metaheuristics. Transportation Science 39 (1), 119–139.

Bullnheimer, B., Hartl, RF., Strauss, C., 1997. Applying the ant system to the vehicle routing problem. In: Second Metaheuristics International Conference, MIC'97, Sophia-Antipolis, France.

Bullnheimer, B., Hartl, RF., Strauss, C., 1999. An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research 89 (0), 319–328.

Chao, I.M., Golden, B.L., Wasil, E., 1995. An improved heuristic for the period vehicle-routing problem. Networks 26 (1), 25–44.

Chen, C.H., Ting, C.J., 2006. An improved ant colony system algorithm for the vehicle routing problem. Journal of the Chinese Institute of Industrial Engineers 23 (2), 115–126.

Croes, G.A., 1958. A method for solving traveling-salesman problems. Operations Research 6 (6), 791–812.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Mans, and Cybernetics 26 (1), 29–41.

Drummond, L.M.A., Ochi, L.S., Vianna, D.S., 1999. An asynchronous parallel metaheuristic for the period vehicle routing problem. Future Generation Computer Systems 17 (4), 379–386.

Francis, P., Smilowitz, K., Tzur, M., 2006. The period vehicle routing problem with service choice. Transportation Science 40 (4), 439–454.

Francis, P., Smilowitz, K., Tzur, M., 2008. The Period Vehicle Routing Problem and its Extensions. The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, New York.

Gong, W., Liu, X., Zhang, J., Fu, Z. 2007. Two-generation ant colony system for vehicle routing problem with time windows. In: 2007 International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2007), Shanghai, vol. 5, pp. 1917–1920.

Hadjiconstantinou, E., Baldacci, R., 1998. A multi-depot period vehicle routing problem arising in the utilities sector. Journal of the Operational Research Society 49 (12), 1239–1248.

Laporte, G., 2009. Fifty years of vehicle routing. Transportation Science 43 (4), 408–416.

Liu, S., Huang, W., Ma, H., 2009. An effective genetic algorithm for the fleet size and mix vehicle routing problems. Transportation Research Part E 45 (3), 434–445.

Matos, A.C., Oliverira, R.C., 2004. An experimental study of the ant colony system for the period vehicle routing problem. Lecture Notes in Computer Science 3172, 286–293.

Paletta, G., 1992. A multiperiod traveling salesman problem: heuristic algorithms. Computers and Operations Research 19 (8), 789–795.

Paletta, G., 2002. The period traveling salesman problem: a new heuristic algorithm. Computers and Operations Research 29 (0), 1343–1352.

Qi, C., Sun, Y., 2008. An improved ant colony algorithm for VRPTW. In: 2008 International Conference on Computer Science and Software Engineering, Wuhan, Hubei, pp. 455-458.

Russell, R.A., Gribbin, D., 1991. A multiphase approach to the period routing problem. Networks 21 (7), 747–765.

Russell, R., Igo, W., 1979. An assignment routing problem. Networks 9 (1), 1–17.

Schoonderwoerd, R., Holland, O., Bruten, J., Rothkrantz, L., 1997. Ant-based load balancing in telecommunications networks. Adaptive Behavior 5 (2), 169–207.

Stützle, T., Hoos, H.H., 2000. MAX–MIN ant system. Future Generation Computer Systems 16 (8), 889–914.

Tan, X., Zhuo, X., Zhang, J., 2005. Ant colony system for optimizing vehicle routing problem with time windows (VRPTW). In: International Conference on Intelligent Computing (ICIC 2006) Kunming Yunan, vol. 2, pp. 209–215.

Yu, B., Yang, Z.Z., Yao, B.Z., 2009. An improved ant colony optimization for vehicle routing problem. European Journal of Operational Research 196 (1), 171–176.

Yu, B., Yang, Z.Z., Xie, J.X., in press. A parallel improved ant colony optimization for multi-depot vehicle routing problem. Journal of the Operational Research Society, doi:10.1057/jors.2009.161.

Zhu, Y., Zhen, T., 2009. Hybrid ant colony algorithm based on vehicle routing problem with time windows. In: 2009 WASE International Conference on Information Engineering Taiyuan Shanxi, vol. 2, pp. 50–53.