**Journal of the Operational Research Society (2011) 62**, 183–188 

www.palgrave-journals.com/jors/

# A parallel improved ant colony optimization for multi-depot vehicle routing problem

B Yu[1], Z-Z Yang[1]* and J-X Xie[2]

[1]*Transportation Management College, Dalian Maritime University, Dalian, P.R. China; and* [2]*College of Mechanical and Electronic Engineering, Hebei Agricultural University, Hebei, P.R. China*

This paper presents a method for solving multi-depot vehicle routing problem (MDVRP). First, a virtual central depot is added to transfer MDVRP to the multi-depot vehicle routing problem with the virtual central depot (V-MDVRP), which is similar to a vehicle routing problem (VRP) with the virtual central depot as the origin. An improved ant colony optimization with coarse-grain parallel strategy, ant-weight strategy and mutation operation, is presented for the V-MDVRP. The computational results for 23 benchmark problems are reported and compared to those of other ant colony optimizations.

*Journal of the Operational Research Society* (2011) **62,** 183–188. doi:10.1057/jors.2009.161
Published online 10 February 2010

**Keywords:** multi-depot vehicle routing problem; virtual central depot; ant colony optimization; ant-weight; mutation operation

## 1. Introduction

Vehicle routing problem (VRP) holds an important place in the logistics field, which includes simultaneously determining routes for several vehicles from a depot to a number of customers without exceeding vehicle capacity constraints. Another well-known generalization of VRP is the multi-depot vehicle routing problem (MDVRP), in which the fleet of vehicles now must serve several depots instead of only one and each vehicle must start from and end at the same depot. Its objective is to find a set of minimum cost routes that can facilitate delivery from multiple depots to a number of customer locations. MDVRP is a NP-hard problem that is very difficult to solve for exact methods like dynamic programming and branch and bound. Thus, many researchers have used heuristic approaches to solve the problem. Early heuristics were developed by Tillman and Cain (1972); Wren and Holliday (1972) and Raft (1982) who used simple construction and some improvement. Better heuristic algorithms were then described by Chao *et al* (1993); Renaud *et al* (1996); Cordeau *et al* (1997); Lim and Wang (2005); Mingozzi (2005) and Crevier *et al* (2007).

MDVRP is a generalization of the standard VRP, in which there are multiple depots. As delivery routes consist of combinations of depots and customers, the MDVRP is very difficult to solve even for relatively small-sized instances compared with the VRP with a single depot (SDVRP). Considering many heuristics successfully applied in solving SDVRP, the objective of this paper is to simplify MDVRP to SDVRP and develop an algorithm that is similar to the approach to SDVRP to solve MDVRP. Assume there is a virtual centre depot and all the vehicles have to start from and end at the 'global' depot. Thus, the actual depots can be considered as the accesses of the 'global' depot. Therefore, the MDVRP can be transformed into the SDVRP with the multiple fixed accesses. The other aim in this paper is to develop a heuristic algorithm, a parallel improved ant colony optimization (PIACO), for the MDVRP with the virtual central depot (V-MDVRP). The remainder of this paper is organized as follows. In Section 2, we describe the standard MDVRP and the V-MDVRP. Section 3 presents the PIACO. Some computational results are discussed in Section 4 and lastly, the conclusions are provided in Section 5.

*Correspondence: Z-Z Yang, Transportation Management College, Dalian Maritime University, Dalian, Liaoning 116026, PRC.
E-mail: yangzhongzhen@263.net

## 2. Problem description

### 2.1. Multi-depot vehicle routing problem

A standard MDVRP can be described as the problem of designing least cost routes from the $H$ depots to a set of geographically scattered points. Each route starts from and ends at the same depot, each customer is visited exactly once by a vehicle and the total demand of each route does not exceed the vehicle capacity $V_h$. An example of the MDVRP is presented in Figure 1. As shown, the stars denote the actual depots and the circles denote the customers. $C$ is the vertex set, $L$ is the edge set and the cost matrix $D=\{d_{ij}, c_i, c_j \in C\}$ corresponding to the distance is defined on $L$. The vertex set $C$ is partitioned into two subsets $C_d = \{c_1, \ldots, c_H\}$ and $C_c = \{c_{H+1}, \ldots, v_{H+N}\}$, respectively, the set of central depots and the set of customers. Each customer $c_i \in C_c$ is associated with a non-negative demand $q_i$ to be delivered and each

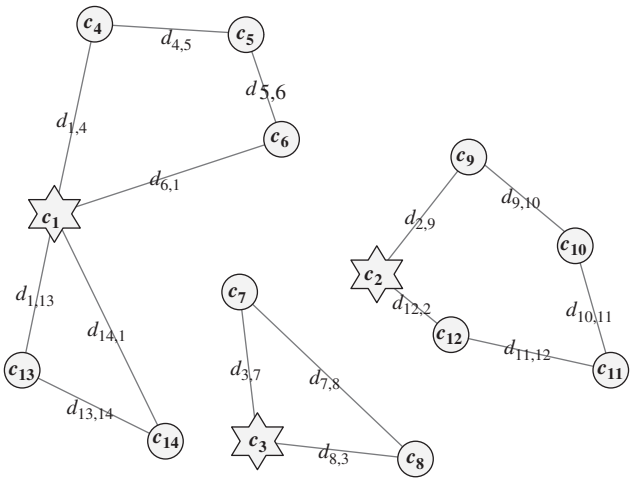


**Figure 1** An example of MDVRP.

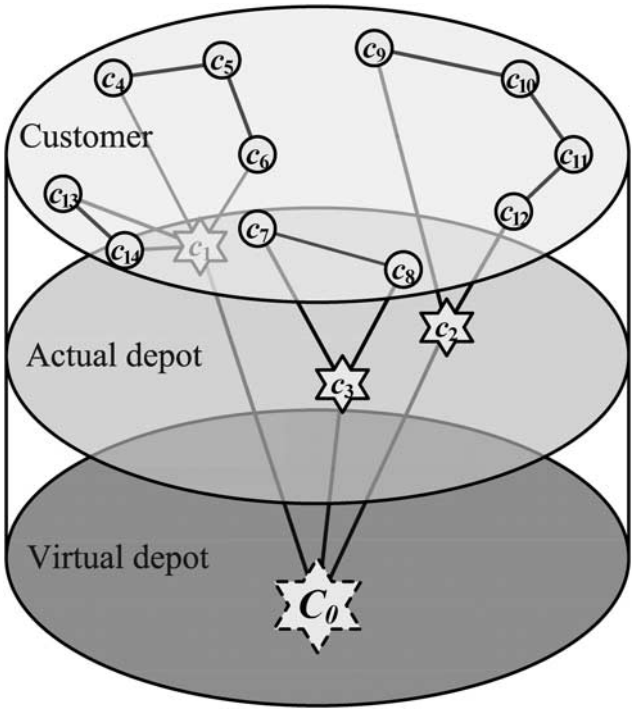


**Figure 2** An example of V-MDVRP.

depot $c_h \in C_d$ is associated with a demand $q_h = 0$ To avoid a vehicle passing two or more depots, we assume $d_{kl} = \infty$ $(c_k, c_l \in C_d)$.

### 2.2. MDVRP with virtual depot

First, a virtual central depot $c_0$, whose distances to the $H$ actual depots and $N$ customers are assumed as 0 and $\infty$, respectively, is added. Each vehicle starts its route from the virtual central depot through one actual depot to customers and returns through the same actual depot to the virtual depot without exceeding the capacity constraints of each vehicle. Thus, the problem is transferred into V-MDVRP, which is similar to a VRP with the virtual central depot as the origin. Figure 2 depicts an example of V-MDVRP with the same customers and actual depots from Figure 1. In addition, all other constraints from the standard MDVRP still apply.

## 3. Parallel improved ACO for V-MDVRP

Ant colony optimization (ACO) is a relatively new optimization method proposed by Dorigo *et al* (1996), which simulates food-seeking behaviours of ant colonies in nature. It has been successfully applied to some classic compounding optimization problems, for example travelling salesman (Dorigo *et al*, 1996; Stützle and Hoos, 2000), vehicle routing (Bullnheimer *et al*, 1999; Bell and McMullen, 2004; Yu *et al*, 2009), telecommunication routing (Schoonderwoerd *et al*, 1997), and product design (Albritton and McMullen, 2007), etc.

The solution of the V-MDVRP is to find a set of minimum cost routes in order to facilitate delivery from the virtual central depot through the $H$ actual depots to a number of customer locations. This is very similar to food-seeking behaviours of ant colonies in nature. If we take the virtual central depot as the *nest*, take the actual depots as the *entries* of *nest*, and take customers as the *food*, solving V-MDVRP can be described as the searching process for *food* starting from the *nest* through an *entry*.

### 3.1. Improved ACO

The specific steps of the improved ACO are as follows:

*Step* 1: *Generation of solutions* In ACO for VRP, the decision-making about combining customers is based on a probabilistic rule taking into account of both the visibility and the pheromone information on an edge. Thus, to select the next customer $j$, the ant uses the following probabilistic formula.

$$p(i,j) = \begin{cases} \dfrac{(\tau(i,j))^{\alpha} \times (\eta(i,j))^{\beta}}{\sum_{l \notin tabu} (\tau(i,l))^{\alpha} \times (\eta(i,l))^{\beta}} & j \notin tabu \\ 0 & otherwise \end{cases} \quad (1)$$

where, $p(i, j) =$ the probability of choosing to combine customers $i$ and $j$ on the route; $\tau(i, j) =$ the pheromone concentration on the edge $(i, j)$. It can tell us how good the combination of these two customers $i$ and $j$ was in previous iterations; $\eta(i, j) =$ the visibility on the edge $(i, j)$. $\alpha$ and $\beta =$ the relative influence of the pheromone trails and the visibility values, respectively; $tabu =$ the set of the infeasible nodes.

*Step* 2: *Mutation operation* In a genetic algorithm, the mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next generation. Furthermore, the study conducted by

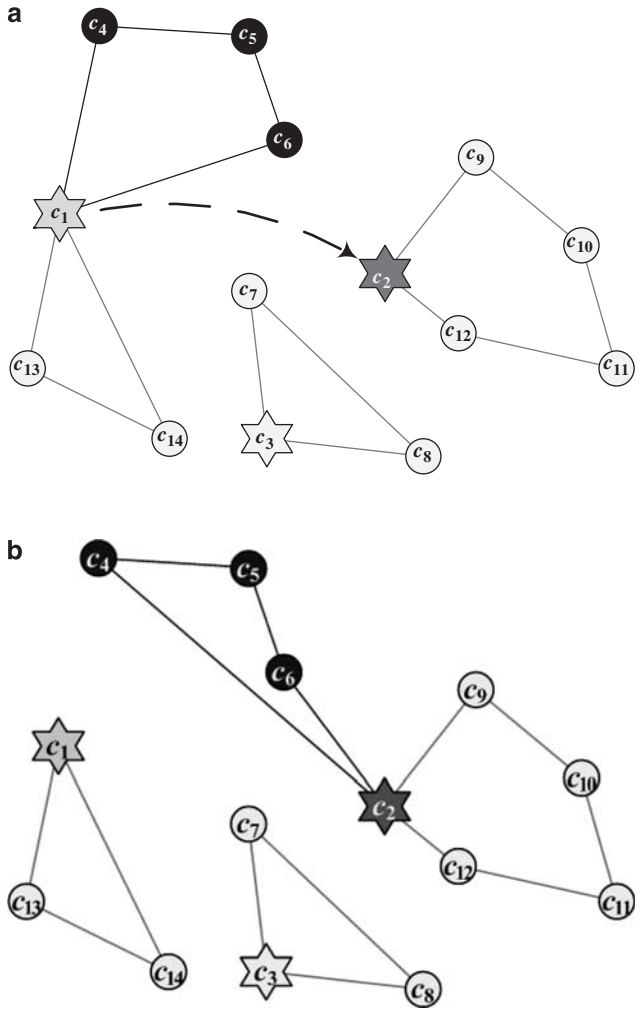


**Figure 3** Illustration of the mutation operation changing depots: (a) Selecting the changed depot and the tour, (b) Changing the depot.

Yu *et al* (2009) has empirically shown that the mutation operation can help the ACO to reach further solutions in the search space. In this paper, there are two mutation operations that produce a new solution that is not very far from the original one. One of the mutation operations is to change the depot on a route and the other is to change a customer on a route. Figure 3 shows two examples of the mutation operations. In the depot mutation, a depot and a tour among all tours through the depot should firstly be selected, for example $c_1$ and $c_1 \rightarrow c_4 \rightarrow c_5 \rightarrow c_6 \rightarrow c_1$ (Figure 3(a)). Then, a new depot is selected among all depots, for example $c_2$ (Figure 3(a)), and a new solution is produced by changing the first selected depot to the second selected depot (Figure 3(b)). Like changing the depot, in the customer mutation, a customer among all customers is firstly selected, for example $c_9$ (Figure 4(a)). The two edges that are linked to the selected customer are deleted and the other customer(s) or the depot that links to the deleted edges is directly linked. Then, a tour among all tours through

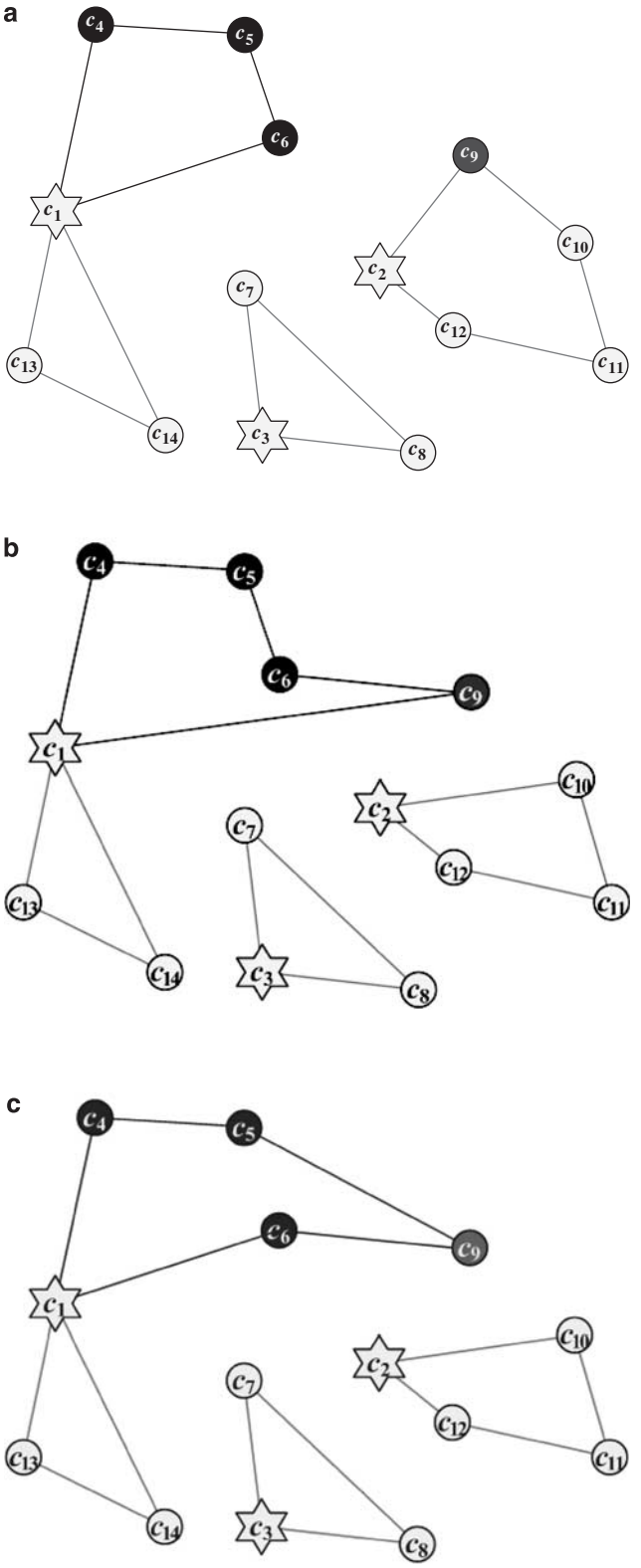


**Figure 4** Illustration of the mutation operation changing customers: (a) Selecting the changed customer, (b) Linking the selected customer to the selected tour, (c) Applying the 2-opt exchange to the mutated tour.

any depot is selected for example $c_1 \to c_4 \to c_5 \to c_6 \to c_1$ (Figure 4(a)). A new solution is produced by linking the selected customer to the last customer and the depot through which the selected tour goes (Figure 4(b)). As the mutation operations may violate vehicle capacity constraints, an approach is to try to fix the resultant capacity violations by adding the tour amounts. Furthermore, the new selected customer in the mutation is directly inserted into the segment between the last customer and the depot, and this can possibly break the local optimality of the new route, like Figure 4(b). Thus, the 2-opt exchange that searches the overall customer sequence by testing all possible pairwise exchanges of the customer locations in a route is used to ensure the local optimality of each route in the newsolution. Then, for this example, the mutated tour can be represented as Figure 4(c) if the new tour does not violate the vehicle capacity constraint.

*Step* 3: *Update of Pheromone Information* The updating of pheromone trails is a key element in ACO, which reflects the quality of the solutions and improves subsequent solutions. First, in order to simulate the natural evaporation of pheromone, the amount of the pheromone on each segment is reduced. Then, the pheromone increments are assigned to each visited edge. This is done with the following pheromone updating equation,

$$\tau(i, j) = \rho \times \tau(i, j) + \Delta\tau(i, j) \quad \rho \in (0, 1) \tag{2}$$

$$\Delta\tau(i, j) = \sum_h \sum_k \Delta\tau_k^h(i, j) \tag{3}$$

where, $\Delta\tau(i, j)$ = the sum of the pheromone increments on edge $(i, j)$; $\Delta\tau_k^h(i, j)$ = the pheromone increments on the $k$th route through the $h$th actual depot on edge $(i, j)$; $\rho$ = the parameter that controls the speed of evaporation; $k$ = the No. of the route.

There are some strategies updating the pheromone increments. In this paper, the ant-weight strategy (Yu *et al*, 2009) is used to assign the increased pheromone, which takes into account of both global and local information. Specifically, the pheromone increment is calculated:

$$\Delta\tau_k^h(i, j) = \begin{cases} \dfrac{Q}{L} \times \dfrac{L - L^h}{(H-1) \times L} \times \dfrac{L^h - f_k^h}{(n^h - 1) \times L^h} & \text{if link } (i, j) \text{ on the } k\text{th route through the } h\text{th actual depot} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where, $Q$ = a constant; $L$ = the total length of all routes in the solution, that is $L = \sum_h L^h$; $L^h$ = the total length of all routes through the $h$th actual depot in the solution, that is $L^h = \sum_k f_k^h$; $f_k^h$ = the length of the $k$th route through the $h$th actual depot in the solution; $n^h$ = the number of the routes through the $h$th actual depot.

| | | $L^h$ | $f_k^h$ | $n^h$ |
|---|---|---|---|---|
| Actual depot | 1 | $L^1 = f_1^1 + f_2^1$ | $f_1^1 = d_{1,4} + d_{4,5} + d_{5,6} + d_{6,1}$<br>$f_2^1 = d_{1,13} + d_{13,14} + d_{14,1}$ | 2 |
| | 2 | $L^2 = f_1^2$ | $f_1^2 = d_{2,9} + d_{9,10} + d_{10,11} + d_{11,12} + d_{12,2}$ | 1 |
| | 3 | $L^3 = f_1^3$ | $f_1^3 = d_{3,7} + d_{7,8} + d_{8,3}$ | 1 |
| $\Sigma$ | | $L = L^1 + L^2 + L^3$ | | $H = 3$ |

**Figure 5** Parameters for updating the increased pheromone.

In the ant-weight strategy, the total pheromone increments on all edges that the solution covers are equal to $Q/L$. The proportion of the pheromone increments shared by the routes through the $h$th actual depot to the total pheromone increments equals $\frac{L - L^h}{(H-1) \times L}$. Similarly, the pheromone increments on each edge on a tour through the $h$th actual depot are related to the contribution of the edge to the tour, that is the pheromone increment proportion of edge $(i, j)$ equal to $\frac{L^h - f_k^h}{(n^h - 1) \times L^h}$. For updating the increased pheromone, the parameters of the example from Figure 1 are as Figure 5. In this way, the valid information obtained from the previous search can be retained for further and more careful search in a more favourable area, which helps speed up the convergence of the algorithm.

### 3.2. Parallelization strategies for IACO

A huge amount of calculation is generated from implementing the ACO in practice, thus parallel implementation of ACO exerts an important role. Referring to parallel genetic algorithms (Eklund, 2004; Solar *et al*, 2002; Yang *et al*, 2007; Yu and Yang, 2009; Yu *et al*, 2007), there are four kinds of parallel strategy for ACO: independent strategies, master-slave strategies, fine-grain strategies and coarse-grain strategies. As the improved ACO proposed in this paper will be implemented in a distributed environment, the coarse-grain strategy is selected in which it requires little communication and can speed up the convergence while ensuring the quality of the solution. The coarse-grain strategy runs several subcolonies in parallel, among which the information exchange is done at certain intervals (*epoch*) or numbers ($n_m$) of iterations (Bullnheimer *et al*, 1998; Middendorf *et al*, 2002). The information exchange employs the ring topology. By exchanging the 'outstanding ants' between sub-colonies, the selection spaces of the subcolonies are diversified to effectively prevent the premature convergence.

## 4. Numerical analysis

The 23 problems from Christofides and Eilon (1969), Gillett and Johnson (1976) and Chao *et al* (1993) were used to validate the performance of the proposed PIACO. The main characteristics of these test problems are summarized in Table 1.

The heuristic described in the previous sections is coded in Visual C++. Net 2003 and executed on a cluster, which consists of eight computers equipped with 512 MB of RAM and a Pentium processor running at 3000 MHz. The parameters in the PIACO (Table 2) are estimated through simulation.

**Table 1** Characteristics of test problems

| *Problem* | *H* | *n* | *Q* | *Problem* | *H* | *n* | *Q* |
|---|---|---|---|---|---|---|---|
| 1 | 4 | 50 | 80 | 13 | 2 | 80 | 60 |
| 2 | 4 | 50 | 160 | 14 | 2 | 80 | 60 |
| 3 | 5 | 75 | 140 | 15 | 4 | 160 | 60 |
| 4 | 2 | 100 | 100 | 16 | 4 | 160 | 60 |
| 5 | 2 | 100 | 200 | 17 | 4 | 160 | 60 |
| 6 | 3 | 100 | 100 | 18 | 6 | 240 | 60 |
| 7 | 4 | 100 | 100 | 19 | 6 | 240 | 60 |
| 8 | 2 | 249 | 500 | 20 | 6 | 240 | 60 |
| 9 | 3 | 249 | 500 | 21 | 9 | 360 | 60 |
| 10 | 4 | 249 | 500 | 22 | 9 | 360 | 60 |
| 11 | 5 | 249 | 500 | 23 | 9 | 360 | 60 |
| 12 | 2 | 80 | 60 | | | | |

After the above work, we continue experimenting 20 times. The proposed PIACO is compared with FIND algorithm (Renaud *et al*, 1996), CGL method (Cordeau *et al*, 1997), a standard ACO (Dorigo *et al*, 1996) and ACO with the ant-weight strategy and the mutation operation (ACO-WM) (Yu *et al*, 2009). These instances and the best known solutions are available at http://neo.lcc.uma.es/radi-aeb/WebVRP//index.html ? / Problem _ Instances / MDVRPInstances.html. Table 3 shows the results obtained using proposed heuristic and the results from other heuristic methods. The numbers in bold are the best solutions among several algorithms. Columns 7–9 present the results from PIACO including the best solutions, the average solutions and the inferior. The results reveal that the PIACO used in this research is able to find the best-known solutions in 15 problems (problem 1–7, 10–13, 16–17, 19 and 21) among the 23 problems. Also, the method can generate competitive solutions for most problems compared with other methods, except for problem 18, 20 and 22. In addition, the PIACO can generate solutions within less than 3% of the optimum solution in most cases. This means our algorithm has a good performance for MDVRP.

**Table 2** Parameters in PIACO

| $m$ *the number of subcolonies* | $p$ *the population of subcolony* | $\alpha$ *a constant* | $\beta$ *a constant* | $Q$ *a constant* | *epoch migrating intervals* | $n_m$ *the number of migrating ants* |
|---|---|---|---|---|---|---|
| 8 | 30 | 2 | 1 | 1000 | 10 | 1 |

**Table 3** The results of PIACO compared with other heuristic methods

| *Problem* | *Best-known solutions* | *FIND* | *CGL* | *ACO* | *ACO-WM* | *PIACO* | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | *Best* | *Mean* | *Inferior %* |
| 1 | **576.86** | **576.86** | **576.86** | **576.86** | **576.86** | **576.86** | 578.54 | 0.29 |
| 2 | **473.53** | **473.53** | **473.87** | **484.28** | **473.53** | **473.53** | 482.09 | 1.81 |
| 3 | **641.18** | **641.18** | 645.15 | 645.16 | **641.18** | **641.18** | 647.62 | 1.00 |
| 4 | **1001.49** | 1003.86 | 1006.66 | 1020.52 | **1001.49** | **1001.49** | 1011.97 | 1.05 |
| 5 | **750.26** | **750.26** | 753.4 | **750.26** | **750.26** | **750.26** | 767.46 | 2.29 |
| 6 | **876.5** | **876.5** | 877.84 | 878.34 | **876.5** | **876.5** | 898.5 | 2.51 |
| 7 | **885.69** | 892.58 | 891.95 | 898.8 | 887.11 | **885.69** | 889.25 | 0.40 |
| 8 | **4437.58** | 4485.08 | 4482.44 | 4508.14 | 4500.15 | 4482.38 | 4659.62 | 3.95 |
| 9 | **3900.13** | 3937.81 | 3920.85 | 4083.44 | 3913 | 3912.23 | 4130.79 | 5.91 |
| 10 | **3663.00** | 3669.38 | 3714.65 | 3747.62 | 3693.4 | **3663.00** | 3749.16 | 2.35 |
| 11 | **3554.08** | 3648.94 | 3580.84 | 3599.93 | 3564.74 | **3554.08** | 3798.31 | 6.87 |
| 12 | **1318.95** | **1318.95** | **1318.95** | 1327 | **1318.95** | **1318.95** | 1330.31 | 0.86 |
| 13 | **1318.95** | **1318.95** | **1318.95** | **1318.95** | **1318.95** | **1318.95** | 1343.73 | 1.88 |
| 14 | **1360.12** | 1365.68 | **1360.12** | 1375.22 | 1373.18 | 1365.68 | 1394.58 | 2.12 |
| 15 | **2505.29** | 2551.45 | 2534.13 | 2588.22 | 2565.67 | 2551.45 | 2603.17 | 2.03 |
| 16 | **2572.23** | **2572.23** | **2572.23** | 2604.9 | **2572.23** | **2572.23** | 2580.42 | 0.32 |
| 17 | **2708.99** | 2731.37 | 2720.23 | 2776.99 | **2708.99** | **2708.99** | 2746.41 | 1.38 |
| 18 | **3702.75** | 3781.03 | 3710.49 | 3907.88 | 3846.05 | ***3781.03*** | 4082.07 | 7.96 |
| 19 | **3827.06** | **3827.06** | **3827.06** | 3863.03 | **3827.06** | **3827.06** | 4017.3 | 4.97 |
| 20 | **4058.00** | 4097.06 | **4058.07** | 4231.28 | 4142 | ***4097.06*** | 4395.7 | 7.29 |
| 21 | **5474.74** | 5656.46 | 5535.99 | 5579.86 | 5495.54 | **5474.74** | 5947.82 | 8.64 |
| 22 | **5702.06** | 5718 | 5716.01 | 5897.64 | 5832.07 | ***5772.23*** | 6196.03 | 7.34 |
| 23 | **6095.36** | 6145.58 | 6139.73 | 6341.61 | 6183.13 | 6125.58 | 6283.54 | 2.58 |

Furthermore, the performance of PIACO is better than the ones of ACO and ACO-WM. This may be because the introduction of the coarse-grain strategy diversifies the ant colony, widens the searching space and prevents the algorithm from trapping in local optimization. Also, the ACO-WM generally gives a better solution than the ACO. This indicates that the ant-weight strategy and the mutation operation are effective compared with ACO. Additionally, the run time not only depends on the CPU of the machines but also on the operation system, the compiler, the programming language and the precision used during the execution of the run. Therefore the comparison of computational effort is not included.

## 5. Conclusions

The MDVRP has been an important problem in the field of the distribution and logistics, which is the generalization of the standard VRP. As the delivery routes consist of combinations of depots and customers, this problem belongs to the class of NP-hard problems. In this paper, by an added virtual central depot, the MDVRP is transferred into V-MDVRP, which is similar to the VRP with the virtual central depot as the origin. Then, a parallel improved ACO is presented, in which three improved strategies: the coarse-grain parallel strategy, the ant-weight strategy and the local search strategy, are applied. The computational results of 23 benchmark problems reveal that our PIACO is effective and efficient.

*Acknowledgements*—This research is financed by the National Science Foundation for Post-doctoral Scientists of China 20080440168 and the Doctoral Program Foundation for Young Scholar of Institutions of Higher Education of China through project 20070151013.

## References

Albritton MD and McMullen PR (2007). Optimal product design using a colony of virtual ants. *Eur J Opl Res* **176**: 498–520.

Bell JE and McMullen PR (2004). Ant colony optimization techniques for the vehicle routing problem. *Adv Eng Inform* **18**: 41–48.

Bullnheimer B, Hartl RF and Strauss C (1999). Applying the ant system to the vehicle routing problem. In: Voss SM, Osman IH and Roucairol C (eds). *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer: Boston, pp 285–296.

Bullnheimer B, Kotsis G and Strauss C (1998). Parallelization strategies for the ant system. In: Leone R, Murli A, Pardalos PM and Toraldo G (eds). *High Performance Algorithms and Software in Nonlinear Optimization*. Kluwer: Dordrecht, pp 87–100.

Chao MI, Golden BL and Wasil EA (1993). A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. *Am J Math Mngt Sci* **13**: 371–406.

Christofides N and Eilon S (1969). An algorithm for the vehicle-dispatching problem. *Opl Res Q* **20**: 309–318.

Cordeau JF, Gendreau M and Laporte G (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* **30**: 105–119.

Crevier B, Cordeau JF and Laporte G (2007). The multi-depot vehicle routing problem with inter-depot routes. *Eur J Opl Res* **176**: 756–773.

Dorigo M, Maniezzo V and Colorni A (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE T Syst Man Cyb* **26**(1): 29–41.

Eklund SE (2004). A massively parallel architecture for distributed genetic algorithms. *Parallel Comput* **30**: 647–676.

Gillett BE and Johnson JG (1976). Multi-terminal vehicle-dispatch algorithm. *Omega* **4**: 711–718.

Lim A and Wang F (2005). Multi-depot vehicle routing problem: A one-stage approach. *IEEE T Automat Sci Eng* **2**: 397–402.

Middendorf M, Reischle F and Schmeck H (2002). Multi colony ant algorithms. *J Heuristics* **8**: 305–320.

Mingozzi A (2005). The multi-depot periodic vehicle routing problem. In: Zucker JD and Saitta L (eds). *Lecture Notes Artificial Intelligence*, vol. 3607. Springer: Berlin, pp 347–350.

Raft OM (1982). A modular algorithm for an extended vehicle scheduling problem. *Eur J Opl Res* **11**: 67–76.

Renaud J, Laporte G and Boctor FF (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Comput Opns Res* **23**: 229–235.

Schoonderwoerd R, Holland O, Bruten J and Rothkrantz L (1997). Ant-based load balancing in telecommunications networks. *Adapt Behav* **5**: 169–207.

Solar M, Parada V and Urrutia R (2002). A parallel genetic algorithm to solve the set-covering problem. *Comput Opns Res* **29**: 1221–1235.

Stützle T and Hoos HH (2000). MAX–MIN ant system. *Future Gener Comput Syst* **16**: 889–914.

Tillman FA and Cain TM (1972). An upperbound algorithm for the single and multiple terminal delivery problem. *Mngt Sci* **18**: 664–682.

Wren A and Holliday A (1972). Computer scheduling of vehicles from one or more depots to a number of delivery points. *Opl Res Q* **23**: 333–344.

Yang ZZ, Yu B and Cheng CT (2007). A parallel ant colony algorithm for bus network optimization. *Comput-Aided Civ Inf Eng* **22**: 44–55.

Yu B and Yang ZZ (2009). A dynamic holding strategy in public transit systems with real-time information. *Appl Intell* **31**: 69–80.

Yu B, Yang ZZ and Cheng CT (2007). Optimizing the distribution of shopping centers with parallel genetic algorithm. *Eng Appl Artif Int* **20**: 215–223.

Yu B, Yang ZZ and Yao BZ (2009). An improved ant colony optimization for vehicle routing problem. *Eur J Opl Res* **196**: 171–176.

*Received April 2008;*
*accepted August 2009 after four revision*