European Journal of Operational Research 196 (2009) 171–176

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing and Logistics

An improved ant colony optimization for vehicle routing problem

Yu Bin [a,*], Yang Zhong-Zhen [a], Yao Baozhen [b]

[a] College of Transportation and Logistics, Dalian Maritime University, Dalian 116026, PR China
[b] School of Civil Engineering and Architecture, Beijing Jiaotong University, Beijing 100044, PR China

ARTICLE INFO

Article history:
Received 8 August 2007
Accepted 22 February 2008
Available online 8 March 2008

Keywords:
Vehicle routing problem
Improved ant colony optimization
Ant-weight strategy
Mutation operation

ABSTRACT

The vehicle routing problem (VRP), a well-known combinatorial optimization problem, holds a central place in logistics management. This paper proposes an improved ant colony optimization (IACO), which possesses a new strategy to update the increased pheromone, called ant-weight strategy, and a mutation operation, to solve VRP. The computational results for fourteen benchmark problems are reported and compared to those of other metaheuristic approaches.

1. Introduction

Finding efficient vehicle routes is a representative logistics problem which has been studied for the last 40 years. A typical vehicle routing problem (VRP) aims to find a set of tours for several vehicles from a depot to a lot of customers and return to the depot without exceeding the capacity constraints of each vehicle at minimum cost. Since the customer combination is not restricted to the selection of vehicle routes, VRP is considered as a combinatorial optimization problem where the number of feasible solutions for the problem increases exponentially with the number of customers increasing (Bell and McMullen, 2004).

Heuristic algorithms such as simulated annealing (SA) (Chiang and Russell, 1996; Koulamas et al., 1994; Osman, 1993; Tavakkoli-Moghaddam et al., 2006), genetic algorithms (GAs) (Baker and Ayechew, 2003; Osman et al., 2005; Thangiah et al., 1994; Prins, 2004), tabu search (TS) (Gendreau et al., 1999; Semet and Taillard, 1993; Renaud et al., 1996; Brandao and Mercer, 1997; Osman, 1993) and ant colony optimization (Doerner et al., 2002; Reimann et al., 2002; Peng et al., 2005; Mazzeo and Loiseau, 2004; Bullnheimer et al., 1999; Doerner et al., 2004) are widely used for solving the VRP. Among these heuristic algorithms, ant colony optimizations (ACO) are new optimization methods proposed by Italian researchers Dorigo et al. (1996), which simulate the food-seeking behaviors of ant colonies in nature. It has been successfully applied as a solution to some classic compounding optimization problems, e.g. traveling salesman (Dorigo et al., 1996) quadratic assignment (Gambardella et al., 1997), job-shop scheduling (Colorni et al., 1994), telecommunication routing (Schoonderwoerd et al., 1997), etc.

If taking the central depot as the *nest* and customers as the *food*, the VRP is very similar to food-seeking behaviors of ant colonies in nature. This makes the coding of an ant colony optimization for the VRP is simple. Among the earliest studies was that of Bullnheimer et al. (1997) who presented a hybrid ant system algorithm with the 2-opt and the saving algorithm for the VRP. Other researches of the ACOs to the VRP included the work by Bullnheimer et al. (1999), Bell and McMullen (2004), Chen and Ting (2006). In the ACOs, the 2-opt exchange was used as an improvement heuristic within the routes found by individual vehicles and the pheromone updating rules mainly considered the global feature of the solution. This paper proposes an improved ant colony optimization with a new pheromone updating rule that can integrate the global feature and the local feature, a mutation operation and the 2-opt exchange for the VRP. The remainder of the paper is organized as follows. Section 2 presents the mathematical model for VRP. In Section 3, we present the IACO with ant-weight strategy and the mutation operation. Some computational results are discussed in Section 4 and lastly, the conclusions are provided in Section 5.

2. Vehicle routing problem

The VRP is described as a weighted graph $G = (C, L)$ where the nodes are represented by $C = \{c_0, c_1, \ldots, c_N\}$ and the arcs are represented by $L = \{(c_i, c_j)\colon i \neq j\}$. In this graph model, c_0 is the central depot and the other nodes are the N customers to be served. Each node is associated with a fixed quantity q_i of goods to be delivered (a quantity $q_0 = 0$ is associated to the depot c_0). To each arc (c_i, c_j) is associated a value $d_{i,j}$ representing the distance between c_i and c_j.

* Corresponding author. Tel.: +86 411 84726756.
E-mail address: minlfish@yahoo.com.cn (B. Yu).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2008.02.028

Each tour starts from and terminates at the depot c_0, each node c_i must be visited exactly once, and the quantity of goods to be delivered on a route should never exceed the vehicle capacity Q.

3. Improved ACO for VRP

3.1. Generation of solutions

Using ACO whose colony scale is P, an individual ant simulates a vehicle, and its route is constructed by incrementally selecting customers until all customers have been visited. The customers, who were already visited by an ant or violated its capacity constraints, are stored in the infeasible customer list (*tabu*).

The decision making about combining customers is based on a probabilistic rule taking into account both the visibility and the pheromone information. Thus, to select the next customer j for the kth ant at the ith node, the ant uses the following probabilistic formula.

$$p_{ij}(k) = \begin{cases} \frac{\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}}{\sum_{h \notin tabu_k} \tau_{ih}^{\alpha} \times \eta_{ih}^{\beta}} & j \notin tabu_k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $p_{ij}(k)$ is the probability of choosing to combine customers i and j on the route, τ_{ij} the pheromone density of edge (i,j), η_{ij} the visibility of edge (i,j), α and β the relative influence of the pheromone trails and the visibility values, respectively and $tabu_k$ is the set of the infeasible nodes for the kth ant.

3.2. Mutation operation

Mutation operation referring to genetic algorithm (Yu and Yang, 2007; Yu et al., 2007) alters each child at a predefined probability. The operators can help the IACO to reach further solutions in the search space. The idea of the mutation operation is to randomly mutate the tour and hence produce a new solution that is not very far from the original one. In this paper, the mutation operator is designed to conduct customer exchanges in a random fashion. Fig. 2 shows the representation of the parent solution in Fig. 1. The steps for the mutation operation are as follows:

Step 1. Select the two tours from the selected parent solution and select the mutating point(s) from the each mutating tour. Fig. 3a shows the 9th customer in the 3rd tour and the 12th customer in the 4th tour are selected.

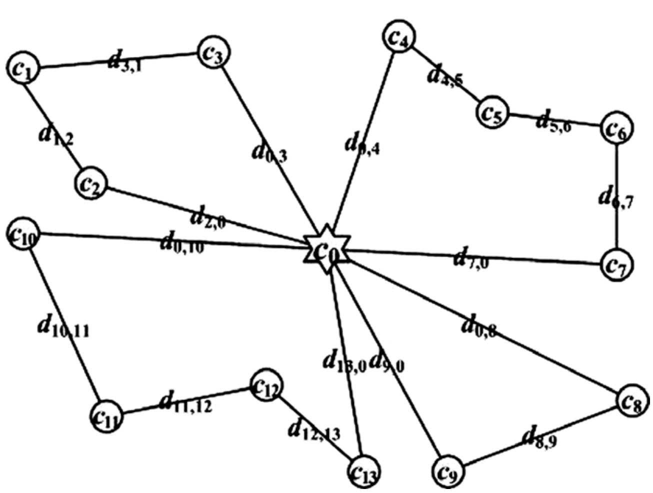

Fig. 1. An example of the VRP.

Parent solution

	Customer			
Route	3	1	2	
	4	5	6	7
	8	9		
	10	11	12	13

Fig. 2. An example of a parent solution.

Step 2. Exchange the customers in the different tours and generate the child solution (see Fig. 3b).

Step 3. Ensure the child solution local optimality. The 2-opt is applied to improve the mutated tours in child solution. Finally, the representation and the tours of the mutated child solution is as Figs. 3c and 4.

However, the mutation operation may violate vehicle capacity constraints. There are two approaches to deal with this situation. The first one is to assign a very high cost for such candidate solutions and accordingly reduce their probability of being selected in the forthcoming search. The second approach is to try to fix the resultant capacity violations by adjusting the delivery amounts. The advantage of the second approach over the first one is that it is more suitable in problems that are more likely to produce vehicle capacity violations and it enables IACO to investigate further points in the search space. Therefore, the second approach is adopted to deal with the vehicle capacity violation situation.

Each route of the solution is mutated with a certain probability p_{m}. Usually, the diversity of the solution is large at the beginning of a run and decreases with the time. We adapt the mutation rate during a run to promote a fast convergence to good solutions during the first generations and to introduce more diversity for escaping from local optima during later stages. The mutation probability at the generation t is

$$p_{\mathrm{m}}(t) = p_{\mathrm{m}}^{\min} + (p_{\mathrm{m}}^{\max} - p_{\mathrm{m}}^{\min})^{1-t/T} \tag{2}$$

where $p_{\mathrm{m}}^{\min}$ and $p_{\mathrm{m}}^{\max}$ are the lower and the upper mutation rates for the beginning and ending, respectively and T and t are the given maximum number of generations and the current generation of the iteration, respectively.

According to preliminary tests, we suggest to set the lower mutation rate to $p_{\mathrm{m}}^{\min} = 1/n_{\mathrm{c}}$, where n_{c} is the number of the customers, and the upper mutation rate to $p_{\mathrm{m}}^{\max} = 1/n_v$, where n_v is the amounts of the routes in the solution.

Parent solution

	Customer			
Route	3	1	2	
	4	5	6	7
	8	9		
	10	11	12	13

Fig. 3a. Procedure of mutation (Step 1).

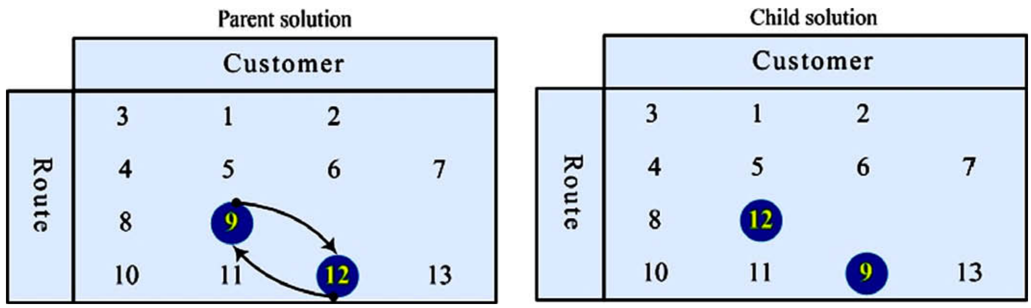

Fig. 3b. Procedure of mutation (Step 2).

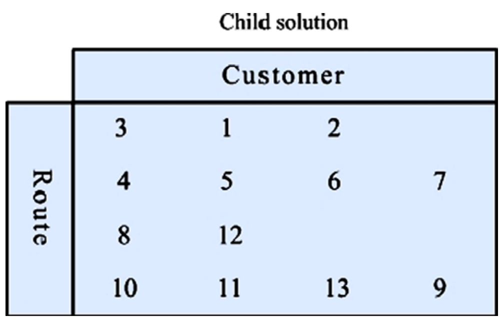

Fig. 3c. Procedure of mutation (Step 3).

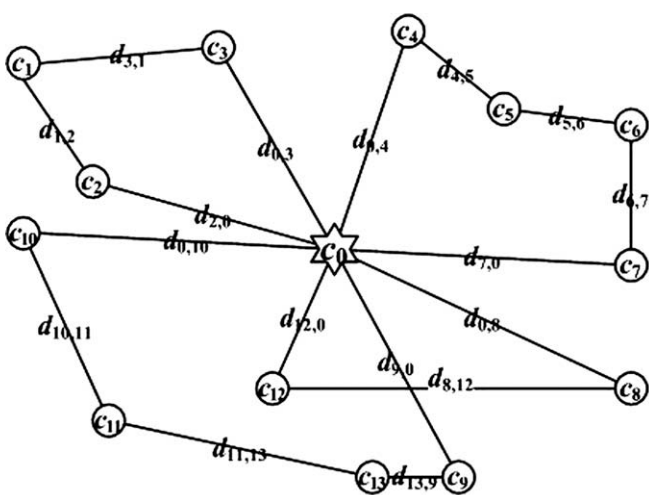

Fig. 4. Tours of the mutated child solution.

3.3. Local search

In the 2-opt exchange, 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.

3.4. 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. First, Pheromone updating is conducted by reducing the amount of pheromone on all links in order to simulate the natural evaporation of the pheromone and to ensure that no one path becomes too dominant. This is done with the following pheromone updating equation,

$$\tau_{ij}^{\text{new}} = \rho \times \tau_{ij}^{\text{old}} + \sum_{k}^{K} \Delta\tau_{ij}^{k} \quad \rho \in (0,1) \tag{3}$$

where τ_{ij}^{new} is the pheromone on the link (i,j) after updating, τ_{ij}^{old} the pheromone on the link (i,j) before updating, ρ the constant that controls the speed of evaporation, k the number of the route, K the number of the routes in the solution and $K > 0$ and $\Delta\tau_{ij}^{k}$ are the increased pheromone on link (i,j) of route k found by the ant.

The pheromone increment updating rule uses the ant-weight strategy presented by Yang et al. (2007). Specifically, the strategy is written as:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{K \times L} \times \frac{D^k - d_{ij}}{m^k \times D^k} & \text{if link } (i,j) \text{ on the kth route} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where Q is a constant, L the total length of all routes in the solution, i.e. $L = \sum_k D^k$, D^k the length of the kth route in the solution, d_{ij} the length of edge (i,j) and m^k the number of customers in the kth routes and $m^k > 0$.

The ant-weight strategy updates the increased pheromone in terms of the solution quality and the contribution of each link to the solution, which consists of two components: the global pheromone increment and the local pheromone increment. In the ant-weight strategy, the quantity of the global pheromone increment, $Q/(K \times L)$, of each route is related to the total length of the solution, while the one of the local pheromone increment $(D^k - d_{ij})/(m^k \times D^k)$ of each link is based on the contribution of link (i,j) to the solution. Since the strategy for updating the increased pheromone considered both the global feature and local one of a solution, it can possibly ensure that the assigned increased pheromone is directly proportional to the quality of routes. The more favorable the link/route is the more pheromone increment is allocated to it, and the more accurate directive information is provided for later search. Meanwhile, by adjusting the pheromone assigning method for the links of current optimal path automatically, the algorithm can facilitate more delicate searches in the next cycle in a more favorable area, which assist in expanding the learning capacity from past searches. The parameters for updating the increased pheromone on the edges in the solution in Fig. 1 are calculated as Fig. 5.

Moreover, in order to prevent from local optimization and increase the probability of obtaining a higher-quality solution, upper and lower limits $[\tau_{\min}, \tau_{\max}]$ are fixed to the updating equation.

$$\tau_{\min} = Q \Big/ \sum_i 2d_{0i}, \tag{5}$$

$$\tau_{\max} = Q \Big/ \sum_i d_{0i}, \tag{6}$$

where d_{0i} is the distance from the central depot to the ith customer.

		Customer			
Route	1	3	1	2	
	2	4	5	6	7
	3	8	9		
	4	10	11	12	13

		D^k	m^k
Route	1	$D^1 = d_{0,3} + d_{3,1} + d_{1,2} + d_{2,0}$	3
	2	$D^2 = d_{0,4} + d_{4,5} + d_{5,6} + d_{6,7} + d_{7,0}$	4
	3	$D^3 = d_{0,8} + d_{8,9} + d_{9,0}$	2
	4	$D^4 = d_{0,10} + d_{10,11} + d_{11,12} + d_{12,13} + d_{13,0}$	4
$\sum$		$L = D^1 + D^2 + D^3 + D^4$	

		$\Delta\tau_{i,j}^k$		
Route	1	•••	$\Delta\tau_{3,1}^1 = \frac{Q}{4 \times L} \times \frac{D^1 - d_{3,1}}{3 \times D^1}$	•••
	2	•••	•••	•••
	3	•••	•••	•••
	4	•••	$\Delta\tau_{10,11}^4 = \frac{Q}{4 \times L} \times \frac{D^4 - d_{10,11}}{4 \times D^4}$	•••
$\sum$			$\sum_{ij} \Delta\tau_{ij}^k = \frac{Q}{4 \times L}$	

Fig. 5. Parameters for updating the increased pheromone.

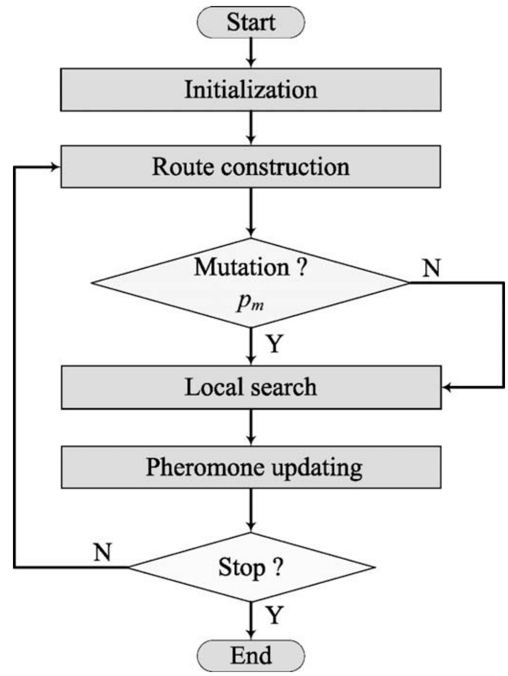

Fig. 6. The flowchart of IACO.

Table 1
Computational results using IACO and the well-known published results

No.	n	Q	Best know	Best	Worst	Average	Time
C1	50	160	524.61[a]	**524.61**	524.61	524.61	2
C2	75	140	835.26[a]	**835.26**	859.3	848.85	11
C3	100	200	826.14[a]	830.00	861.12	844.32	30
C4	150	200	1028.42[a]	**1028.42**	1067.1	1042.52	211
C5	199	200	1291.45[b]	1305.5	1344.41	1321.91	677
C6	50	160	555.43[a]	**555.43**	568.89	560.14	24
C7	75	140	909.68[a]	**909.68**	942.29	919.1	20
C8	100	200	865.94[a]	**865.94**	888.89	871.52	57
C9	150	200	1162.55[a]	**1162.55**	1228.9	1194.87	307
C10	199	200	1395.85[b]	**1395.85**	1433.68	1412.92	840
C11	120	200	1042.11[a]	**1042.11**	1056.26	1048.12	61
C12	100	200	819.56[a]	**819.56**	842.51	823.66	31
C13	120	200	1541.14[a]	1545.93	1572.29	1552.25	127
C14	100	200	866.37[a]	**866.37**	869.12	867.05	43

[a] Taillard (1993).
[b] Rochat and Taillard (1995).

3.5. Overall procedure

The flowchart of our IACO for the VRP is shown in Fig. 6.

4. Numerical analysis

The heuristics described in the previous sections is applied to the 14 vehicle routing problems which can be downloaded from the OR-library (see Beasley, 1990), and which have been widely used as benchmarks, in order to compare its ability to find the solution to VRP. The information of the 14 problems is shown in columns 2–4 in Table 1, which consists of the problem size n, the vehicle capacity Q, and the well-known published results (Taillard, 1993; Rochat and Taillard, 1995). The IACO parameters used for VRP instances are $Q = 1000$, $\alpha = 2$, $\beta = 1$ and $\rho = 0.8$. Then, the IACO were coded in Visual C++.Net 2003 and executed on a PC equipped with 512 MB of RAM and a Pentium processor running at 1000 MHz. Columns 5–8 present the results from IACO including the best solution, the worst solution, and the average solution and average run time (second). The numbers in bold are the results as the best-known solutions.

To evaluate the ant-weight strategy and the mutation operation, the two ant colony optimizations with different strategy are constructed. The first one is a standard ant colony optimization with the ant-weight strategy (denoted by ACO-W) and the other is a standard ant colony optimization with the mutation operation (denoted by ACO-M). The computational results are as shown in Table 2. The numbers in bold are the best solutions among three algorithms. It can be observed that ACO-M can obtain the same solutions as IACO in test problem 1, 2, 3, 6, 7, 11, 12 and 14, while ACO-W can only obtain the optimum solutions in test problem 1, 2, 6, 7 and 14. Compared with ACO-W, the ACO-M generally provides better solutions for the 14 problems. This may be attributed that the introduction of the mutation operation can diversify the ant colony, explore new possible solution space and prevent the algorithm from trapping in local optimization. However, while the mutation operation improves the solutions, it also increases the computation times. We can see that the times consumed by

Table 2
Computational results using IACO, ACO-W and ACO-M

No.	IACO				ACO-W				ACO-M			
	Best	Worst	Average	Time	Best	Worst	Average	Time	Best	Worst	Average	Time
C1	**524.61**	524.61	524.61	2	**524.61**	524.61	524.61	2	**524.61**	524.61	524.61	2
C2	**835.26**	859.3	848.85	11	**838.25**	859.3	850.05	10	**835.26**	859.3	849.33	13
C3	**830.00**	861.12	844.32	30	834.36	861.12	851.02	27	**830.00**	861.12	847.26	51
C4	**1028.42**	1067.1	1042.52	211	1044.89	1087.3	1058.62	198	1033.26	1084.31	1052.52	584
C5	**1305.5**	1344.41	1321.91	677	1335.36	1387.85	1362.37	602	1310.21	1377.29	1340.07	1,134
C6	**555.43**	568.89	560.14	24	**555.43**	571.17	563.32	24	**555.43**	577.49	564.18	24
C7	**909.68**	942.29	919.1	20	**909.68**	947.87	927.06	19	**909.68**	950.1	931.07	22
C8	**865.94**	888.89	871.52	57	876.52	901.06	888.87	52	869.91	901.06	880.03	79
C9	**1162.55**	1228.9	1194.87	307	1204.47	1279.39	1252.05	292	1188	1253.31	1222.24	772
C10	**1395.85**	1433.68	1412.92	840	1439.07	1520.04	1487.78	780	1412.12	1492.36	1466.62	1,320
C11	**1042.11**	1056.26	1048.12	61	1051.71	1077.33	1059.13	55	**1042.11**	1060.61	1055.52	204
C12	**819.56**	842.51	823.66	31	833.31	850.04	840.61	30	**819.56**	851.11	844.06	77
C13	**1545.93**	1572.29	1552.25	127	1571.05	1622.88	1592.83	118	1556.86	1618.58	1588.21	402
C14	**866.37**	869.12	867.05	43	**866.37**	870.18	868.09	38	**866.37**	870.18	868.61	79

ACO-M are more than ACO-W. This may be because that in ACO-W, the ant-weight strategy is used for updating the increased pheromone. It can assign increased pheromone according to the quality the solution. This improves the learning capacity of the algorithm from past searches, and enhances the efficiency. Furthermore, IACO integrated the ant-weight and the mutation operation can provide the best solutions and consume the less computation times compared with ACO-W and ACO-M.

Table 3
Deviations from the best known solution of several metaheuristic approaches

Prob.	RR-PTS	G-TS	OSM-TS	OSM-SA	B-AS	IACO
C1	**0.00**	**0.00**	**0.00**	0.65	**0.00**	**0.00**
C2	0.01	0.06	1.05	0.40	1.08	**0.00**
C3	**0.17**	0.40	1.44	0.37	0.75	0.47
C4	1.55	0.75	1.55	2.88	3.22	**0.00**
C5	3.34	2.42	3.31	6.55	4.03	**1.08**
C6	**0.00**	**0.00**	**0.00**	**0.00**	0.87	**0.00**
C7	**0.00**	0.39	0.15	**0.00**	0.72	**0.00**
C8	0.09	0.00	1.39	0.09	0.09	**0.00**
C9	0.14	1.31	1.85	0.14	2.88	**0.00**
C10	1.79	1.62	3.23	1.58	4.00	**0.00**
C11	**0.00**	3.01	0.09	12.85	2.22	**0.00**
C12	**0.00**	**0.00**	0.01	0.79	**0.00**	**0.00**
C13	0.59	2.12	0.31	**0.31**	1.22	**0.31**
C14	**0.00**	**0.00**	**0.00**	2.73	0.08	**0.00**
Average	0.55	0.86	1.03	2.10	1.51	0.14

Our IACO are compared with five other meta-heuristic approaches in the paper proposed by Bullnheimer et al. (1997), which consisted of parallel tabu search algorithm (RR-PTS) by Rego and Roucairol, a tabu search algorithm (G-TS) by Gendreau et al., tabu search (OSM-TS), a simulated annealing algorithm (OSM-SA) by Osman and ant system algorithm (B-AS) by Bullnheimer et al. (1997). The comparison of the deviations from the best known solution is shown in Table 3. The performance of our IACO is best among all meta-heuristic approaches, who produces in eleven problems of fourteen test problems and yields the lowest average deviation. Also, compared with OSM-SA, the tabu search approaches are able to provide better solutions. Also, compared OSM-SA, the tabu search approaches can provide better solutions.

For a correct evaluation and comparison of the quality of six algorithms the computing times must be taken into account. However, a correct evaluation and comparison of the computing times is generally tough due to the enormous variety of computers available and used by different researchers. A very rough measure of computers' performance can be obtained using Dongarra's (Dongarra, 2001) tables where the number (in millions) of floating-point operations per second (Mflop/seconds) executed by each computer was used, when solving standard linear equations, with LINPACK program. Regarding computational times, Rego and Roucairol used a sun sparc 4 (about 5.7 MFlop/s), Gendreau et al. used a 36 MHz Silicon Graphics (about 6.7 MFlop/s), Osman used a VAX 8600(about 2.48 MFlop/s), Bullnheimer et al. used a Pentium 100 MHz (about 8 MFlop/s). In this research, the Pentium 1 GHz running IACO has an estimated power of 75 MFlop/s. Table 4 shows

Table 4
Computation times of several metaheuristic approaches

Probability	RR-PTS		G-TS		OSM-TS		OSM-SA		B-AS		IACO
	Run times	Scaled times	Run times	Scaled times	Run times	Scaled times	Run times	Scaled times	Run times	Scaled times	Run times
C1	66	5.0	84	7.5	60	2.0	6	0.2	6	0.6	2
C2	2604	197.9	2352	210.2	48	1.5	3564	117.8	78	8.3	11
C3	1578	119.9	408	36.4	894	29.5	6174	204.1	228	24.3	30
C4	2910	221.2	3270	292.1	1764	58.3	4296	142.1	1104	117.8	211
C5	4626	351.6	5028	449.1	1704	56.3	1374	45.4	5256	560.6	677
C6	144	10.9	468	41.8	60	2.0	696	23.0	6	0.6	24
C7	1236	94.0	1908	170.4	744	24.6	312	10.4	102	10.9	20
C8	1134	86.2	354	31.6	1962	64.8	366	12.1	288	30.7	57
C9	1794	136.3	1278	114.1	2472	81.8	59,016	1951.5	1650	176.0	307
C10	2562	194.8	2646	236.4	4026	133.0	2418	79.9	4908	523.5	840
C11	672	51.1	714	63.8	780	25.8	264	8.7	552	58.9	61
C12	96	7.3	102	9.2	342	11.2	48	1.5	300	32.0	31
C13	120	9.1	2088	186.5	1578	52.1	4572	151.1	660	70.4	127
C14	1482	112.7	1782	159.2	582	19.3	300	9.9	348	37.1	43
Average	–	114.14	–	143.46	–	40.17	–	196.98	–	117.98	172.93

the origin computation times and the scaled computation times, which use Pentium 1 GHz as the baseline, of six approaches.

The performance of IACO is competitive when compared with other meta-heuristic approaches, such as SA, and TS. Although the run times are not favor in IACO, our IACO still seems to be superior in terms of solution quality with an average deviation of 0.14%. Considering the very rough measure, the scaled times are viewed as the assistant aspect of the performance. Regarding the computation efficiency, we find that the IACO can find very good solutions in an acceptable time.

5. Conclusions

The VRP has been an important problem in the field of distribution and logistics. Since the delivery routs consist of any combination of customers, this problem belongs to the class of NP-hard problems. This paper presents an IACO with ant-weight strategy and a mutation operation. The computational results of 14 benchmark problems reveal that the proposed IACO is effective and efficient. Further research on additional modifications of the IACO to extensions of the vehicle routing problem with time windows or with more depots, are of interest.

References

Baker, B.M., Ayechew, M.A., 2003. A genetic algorithm for the vehicle routing problem. Computers & Operations Research 30, 787–800.

Beasley, J.E., 1990. OR-Library: distributing test problems by electronic mail. Journal of the Operational Research Society 41, 1069–1072.

Bell, J.E., McMullen, P.R., 2004. Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics 1 (8), 41–48.

Brandao, J., Mercer, A., 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. European Journal of Operational Research 100, 180–191.

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

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.

Chiang, W.C., Russell, R., 1996. Simulated annealing meta-heuristics for the vehicle routing problem with time windows. Annals of Operations Research 93, 3–27.

Colorni, A., Dorigo, M., Maniezzo, V., Trubian, M., 1994. Ant system for job-shop scheduling. Jorbel–Belgian Journal of Operations Research Statistics and Computer Science 34 (1), 39–53.

Doerner, K.F., Gronalt, M., Hartl, R.F., Reimann, M., Strauss, C., Stummer, M., 2002. Savings ants for the vehicle routing problem. Applications of Evolutionary Computing. Springer, Berlin.

Doerner, K.F., Hartl, R.F., Kiechle, G., Lucka, M., Reimann, M., 2004. Parallel ant systems for the capacitated vehicle routing problem. In: Evolutionary Computation in Combinatorial Optimization: 4th European Conference, EvoCOP 2004, LNCS 3004, pp. 72–83.

Dongarra, J., 2001. Performance of various computer using standard linear equations software. Report CS-89-85, University of Tennessee.

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

Gambardella, L., Taillard, E., Dorigo, M., 1997. Ant Colonies for the QAP, Technical Report 97-4, IDSIA, Lugano, Switzerland.

Gendreau, M., Laporte, G., Musaraganyi, C., Taillard, E.D., 1999. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers & Operations Research 26, 1153–1173.

Koulamas, C., Antony, S., Jaen, R., 1994. A survey of simulated annealing applications to operations research problems. Omega 22 (1), 41–56.

Mazzeo, S., Loiseau, I., 2004. An ant colony algorithm for the capacitated vehicle routing. Electronic Notes in Discrete Mathematics 18, 181–186.

Osman, I.H., 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research 41, 421–451.

Osman, M.S., Abo-Sinna, M.A., Mousa, A.A., 2005. An effective genetic algorithm approach to multiobjective routing problems (morps). Applied Mathematics and Computation 163, 769–781.

Peng, W., Tong, R.F., Tang, M., Dong, J.X., 2005. Ant colony search algorithms for optimal packing problem. ICNC 2005, LNCS 3611, pp. 1229–1238.

Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 31, 1985–2002.

Reimann, M., Stummer, M., Doerner, K., 2002. A savings based ant system for the vehicle routing problem. In: Langdon, W.B. et al. (Eds.), GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann, San Francisco.

Renaud, J., Laporte, G., Boctor, F.F., 1996. A tabu search heuristic for the multi-depot vehicle routing problem. Computers & Operations Research 23 (3), 229–235.

Rochat, Y., Taillard, R.E., 1995. Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics 1, 147–167.

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

Semet, F., Taillard, E.D., 1993. Solving real-life vehicle routing problems efficiently using taboo search. Annals of Operations Research 41, 469–488.

Taillard, R.E., 1993. Parallel iterative search methods for vehicle routing problems. Networks 23, 661–673.

Tavakkoli-Moghaddam, R., Safaei, N., Gholipour, Y., 2006. A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. Applied Mathematics and Computation 176, 445–454.

Thangiah, S.R., Osman, I.H., Sun, T., 1994. Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows. Technical Report 27, Computer Science Department, Slippery Rock University.

Yang, Z.Z., Yu, B., Cheng, C.T., 2007. A parallel ant colony algorithm for bus network optimization. Computer-Aided Civil and Infrastructure Engineering 22, 44–55.

Yu, B., Yang, Z.Z., 2007. A dynamic holding strategy in public transit systems with real-time information. Applied Intelligence, (accepted for publication), doi:10.1007/s10489-007-0112-9.

Yu, B., Yang, Z.Z., Cheng, C.T., 2007. Optimizing the distribution of shopping centers with parallel genetic algorithm. Engineering Applications of Artificial Intelligence 20 (2), 215–223.