Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080658

An Exponential Entropy-based Hybrid Ant Colony Algorithm for Vehicle Routing Optimization

Chengming Qi[1,*] *and Ping Li*[2]

[1] College of Automation, Beijing Union University 100101, Beijing, China
[2] Institute of Logistics, Beijing Union University 100101, Beijing, China

Received: 28 Nov. 2013, Revised: 26 Feb. 2014, Accepted: 27 Feb. 2014
Published online: 1 Nov. 2014

Abstract: Vehicle routing problem(VRP) is important combinatorial optimization problems which have received considerable attention in the last decades. The optimization of vehicle routing problem is a well-known research problem in the logistics distribution. In order to overcome the prematurity of Ant Colony Algorithm (ACA) for logistics distribution routing optimization, a hybrid algorithm combining improved ACA with Iterated Local Search (ILS) is proposed. The proposed algorithm adjusts the pheromone trail to balance the convergence rate and diversification of solutions self-adaptively. The exponential entropy is used to control the path selection and pheromone updating strategy. Combining with ILS is to avoid local best solutions and accelerate the search. Computational results denote the efficiency of the proposed algorithm on some standard benchmark problems.

Keywords: Ant Colony System, Information Entropy, Iterated Local Search, Logistics vehicle Routing

1 Introduction

The optimization of Vehicle Routing Problem (VRP) in the logistics distribution is a well-known research problem. Nowadays, companies acknowledge the importance of a better design and an efficient management of their logistics distribution, for achieving higher quality services at the lowest effort. VRP is a well-known combinatorial optimization problem with considerable economic significance. A typical VRP can be described as the problem of designing routes from one depot to a set of geographically scattered points, such as cities, warehouses, customers with the least effort. VRP has been studied extensively because of the interest in its applications in logistic and supply-chains management.

Biological inspired computation is a field focusing on the development of computational tools modeled following the principles that exist in natural systems. The results presented in recent publications show that bio-inspired approaches are now highly competitive with other state-of-the-art heuristics [1,2].

The Ant System (AS), introduced by Colorni, Dorigo and Maniezzo [3], is a distributed meta-heuristic for hard combinatorial optimization problems. It was first used on solving the Traveling Salesman Problem (TSP). One of the most efficient Ant Colony Optimization (ACO) based implementations is based on Ant Colony System (ACS) [4], which introduced a particular pheromone trail updating procedure for intensifying the search in the neighborhood of the best computed solution. Silvia Mazzeo et al. [5] have improved Capacitated Vehicle Routing Problem (CVRP) by means of an ACO algorithm. Watcharasitthiwat & Wardkein [6] developed an improved ACO for optimization of network topology.

Entropy comes from physics. It is used to describe chaos and disorder. The bigger the entropy value is, the more the confusion degree is. In the information theory, Shannon defined the information entropy as the probability of random event. A larger uncertainty of the variables has more information entropy.

Iterated Local Search (ILS) [7] is a very simple and powerful stochastic local search method that has proved to be one of the best performing approximation algorithms. Prins presented a hybrid metaheuristic method for VRP [8]. The method extends the classical ILS. The result shows that ILS is able to create efficient and fast algorithms for the VRP.

In this paper, we propose an improved Ant Colony Algorithm (ACA), named E2ACA (Exponential Entropy-Based Ant Colony Algorithm), to overcome the

* Corresponding author e-mail: zdhtchengming@buu.edu.cn

problem of premature convergence of ACA for logistics distribution routing optimization. The information entropy is used to control the path selection and the pheromone updating strategy. Self-adaptively adjustment mechanism of pheromone adopted in our algorithm can balance the speed of convergence and diversity of solutions. Combining with ILS is to avoid local-best solutions and accelerate the search.

The paper is organized as follows. In Section 2 introduces the VRP and the solution construction mechanism used by the ACA. Section 3 presents the entropy-based hybrid ACA. In Section 4 we provide computational results of E2ACA on a set of benchmark problems. We conclude in Section 5 with a brief summary of the contributions of this paper.

2 Problem formulation

In this section, we will introduce VRP and apply ant colony algorithm to the VRP.

2.1 Vehicle Routing Problem

Vehicle routing problem is a combinatorial optimization and integer programming problem, which with a fleet of vehicles aims at investigating how to service a number of customers. It is a fundamental research problem in the fields of logistics and transportation, and has attracted the attention of a large number of researchers. The vehicle routing problem was first introduced by Dantzig and Ramser [9]. As an NP-hard problem, a number of approximation techniques were proposed to solve it. In the last 15 years, Meta-heuristic algorithms, such as Tabu Search [10,11,12], Simulated annealing [13], Variable Neighborhood Search [14], self adaptive local search [15] and Greedy Randomized Adaptive Search Procedure [8, 16] are also applied efficiently in the VRP.

Recently, a number of nature inspired metaheuristic algorithms have been applied for the VRP. They are genetic algorithms [17,18,19], ant colony optimization [20], particle swarm optimization [21].

The capacitated vehicle routing problem is the basic version of the VRP. The name derives from the constraint of having vehicles with limited capacity. The capacitated vehicle routing problem is NP-hard, a CVRP is more difficult to solve than a TSP because it contains one or more TSP as subproblems.

The classic capacitated vehicle routing problem can be described as follows: N customers geographically dispersed in a planar region must be served from a unique depot. Each customer asks for a quantity $q_i(i=1,2,\cdots,n)$ of goods.The transport cost from node i to node j is c_{ij}. m vehicles with a fixed capacity Q are available to deliver the goods stored in the depot. Each customer must be visited just once by only one vehicle. The objective of the problem is minimizing the total cost of all routes without violating the individual capacity of each vehicle. The depot is denoted by i=*0*. The model can be written as follows:

$$min \sum_{k=1}^{m} \sum_{j=0,j\neq i}^{n} \sum_{i=0}^{n} c_{ij}x_{ij}^{k} \quad (1)$$

$$x_{ij}^{k} = \begin{cases} 1 & \text{if vehicle } k \text{ goes from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}$$

$$s.t. \sum_{k=1}^{m} \sum_{j=1}^{n} x_{ij}^{k} \leq m \qquad i=0 \quad (2)$$

$$\sum_{j=1}^{n} x_{ij}^{k} = \sum_{j=1}^{n} x_{ji}^{k} \leq 1 \qquad i=0, k\in\{1,2,\cdots,m\} \quad (3)$$

$$\sum_{k=1}^{m} \sum_{j=0,j\neq i}^{n} x_{ij}^{k} = 1 \qquad i\in\{1,2,\cdots,n\} \quad (4)$$

$$\sum_{k=1}^{m} \sum_{i=0,i\neq j}^{n} x_{ij}^{k} = 1 \qquad j\in\{1,2,\cdots,n\} \quad (5)$$

$$\sum_{i=1}^{n} d_i \sum_{j=0,j\neq i}^{n} x_{ij}^{k} \leq Q \qquad j\in\{1,2,\cdots,m\} \quad (6)$$

The objective function (1) is minimizing the total distance traveled. Constraint (2) assures the number of vehicles originating from the depot is not more than m. Constraint (3) states that each of the k vehicles has to leave and go back to the depot. Constraints (4) and (5) assure that each customer is visited exactly once. Constraint (6) is the capacity constraints.

2.2 Applying ant colony algorithm to the VRP

In order to solve the VRP, the artificial ants choose cities to visit to construct routes, continuing until each customer has been served. A new route will be started from the depot when all remaining choices have been visited.

At each step, every ant k computes a set of feasible expansions to its current partial solution and probabilistically selects one of these, according to a probability specified distribution as follows. For ant k the probability p_{ij}^{k} of visiting customer j after customer i, the last visited customer, depends on the combination of two values [22]:the attractiveness η_{ij} of arc *(i,j)*, and the pheromone level τ_{ij} of arc *(i,j)*.

This heuristic uses a population of m agents which construct solutions step by step. The best solution are rewarded when all the ants have constructed their tour.

Construction of vehicle routes. ACO goal is to find a shortest tour. In ACA m ants build tours in parallel, where

m is a parameter. Each ant is randomly assigned to a starting node and build a solution (complete tour). A tour is built node by node: each ant iteratively adds new nodes until all nodes have been visited. When ant k is located in node i, it chooses the next node j probabilistically in the set of feasible nodes N_i^k(i.e., the set of nodes that still have to be visited). The probabilistic rule used to construct a tour is the following: with probability q_0 a node with the highest $[\tau_{ij}]^\alpha[\eta_{ij}]^\beta$, $j \in N_i^k$ is chosen (exploitation), while with probability $(1-q_0)$ the node j is chosen with a probability p_{ij} proportional to $[\tau_{ij}]^\alpha[\eta_{ij}]^\beta$, $j \in N_i^k$ (exploration).

With $\Omega = \{v_j \in V : v_j \text{ is feasible to be visited}\} \cup \{v_0\}$, city v_j is selected to be visited after city v_i according to a *random-proportional rule* [4] that can be stated as follows:

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{h\in\Omega}[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta} & \text{if } v_j \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

This probability distribution is determined by the parameters α and β that exert the relative influence of the trails and the visibility, respectively. For the TSP Dorigo et al. [23] define the visibility as the reciprocal of the distance. The same is done for the VRP in [24] where the probability selected is then further extended by problem specific information. There, the inclusion of savings and capacity utilization both lead to better results. The latter is relative costly in terms of computation time for it is calculated in each step of an iteration.

Pheromone trail update: An ant iteratively constructed a complete solution. After all ants have generated their solutions, according to certain rules, the most suitable solution is gotten. A solution can be improved by applying a local search. In addition, ants are allowed to deposit pheromone, taking into consideration the quality of the solutions they have constructed.

Initially no information is contained in the pheromone trail, meaning that all pheromone trails are equal to a value. Since pheromone trails are updated by taking into account the absolute value of the solution obtained, must take a value that depends on the value of the solutions being visited.

In this paper, we adopt the strategy that dynamically adjusts pheromone trails by limiting the possible trail values between some maximum and minimum limitations $[\tau_{min}, \tau_{max}]$. The update of the pheromone trail is done in a different way than those of the standard model where all the ants update the pheromone trail. Indeed, this manner of updating the pheromone trail results in a very slow convergence of the algorithm [4]. For speeding-up the convergence, we update the pheromone trail by taking into account only the best solution. The update is made according to the following equation:

$$\tau_{ij}(t+1) = \rho^{f(k)}\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k \tag{8}$$

where m is the number of ants; ρ is a coefficient representing the trace's persistence ($(1-\rho)$ represents the evaporation), and $\rho \in [0,1]$. $f(k)$ is the function which is directly proportional to the numbers of improved solution in iterations, such as, $f(k) = k/c$, (c is a constant); τ_{ij}^k is the pheromone trail on route (i,j) constructed by the k-th ant.

Constructive procedure is described briefly as follows.

Step 1. Read the input data;

Step 2.Initialize parameters and pheromone matrix;

Step 3. Construct a VRP subtour solution according to Eq.(7);

Step 4. Put the chosen vertices in the tabu list of the k-th;

Step 5. Repeat Step 3 Step 4 until every ant have visited all vertices, i.e. each ant construct a tour;

Step 6. Select the best tour found solution so far.

Step 7. Update the pheromone matrix Eq.(8).

Step 8. Repeat from Step 3 to Step 7 until a pre-specified stopping criterion is met.

3 Iterated local search and entropy-based hybrid ACA

In this section, we present our hybrid algorithm E2ACA. The approach applies local search algorithm to ACA iteratively. The information entropy is used to control the path selection and the pheromone updating strategy.

3.1 Iterated local search

The iterated local search meta-heuristic works as follows: one builds a sequence of solutions iteratively generated by the heuristic, leading to far better solutions than one used repeated random trials of that heuristic. In order to apply an ILS algorithm, three basic procedures have to be specified. Given the current solution s_0, these is a procedure Perturbation, which perturbs the current solution s leading to the same intermediate solution s', a procedure *LocalSearch* that takes s' to a local optimum s'', and an *AcceptanceCriterion* that decides from which solution the next perturbation step is applied [25]. Each solution established in the former stage s_0 is taken to a local optimum. Then, a local search procedure is applied iteratively from a starting solution obtained by a perturbation of the current search point. The iterated local search procedure is described briefly as follows.

Step 1 (Initialization). Generate some initial solution s_0.

Step 2 (Local search). Let s the best solution: s = LocalSearch(s_0).

Step 3 (Perturbation). Let s' the intermediate solution after Perturbation: s' = Perturbation(s,history), where *history* is previous solution set.

Step 4 (Local search). Run local search on s': s''= LocalSearch(s').

Step 5 (AcceptanceCriterion). AcceptanceCriterion decides from which solution the next perturbation step is applied: s = AcceptanceCriterion(s,s'',history).

Step 6. Repeat from Step 3 to Step 5 until termination condition met.

Step 7. Output s, the best local solution ever found.

In the ant colony algorithms, the path selection is related to the pheromone of each edge, where uncertainty exists. So we introduce the entropy to measure the uncertainty of pheromone in each edge, and use information entropy to adjust the path chosen strategies and pheromone updated strategy. Specifically, for every i-th customer, $i \in [1,m]$ of an ant during the process of constructing a solution we computed the entropy

$$S(t) = -\sum_{j \in D} P_{ij} log P_{ij} \tag{9}$$

where p_{ij} is defined in Eq. (7).

In order to choose the path we can get the information entropy value and determine the degree of certainty. This definition is a combination of its own characteristics of the ant colony algorithm, which combined a sequence of arithmetic with information entropy, to regulate the algorithm adaptively.

Exponential entropy overcame the deficiency of the logarithmic information entropy through improving undefined value and zero problems of logarithmic entropy. In addition, because logarithmic computational speed is slow, exponentiation computation can be greatly reduce computational time.

In this paper, we defined information quantity of an event i with probability p_i as Eq. (10):

$$\triangle I(p_i) = e^{(1-p_i)} \tag{10}$$

exponential entropy is Eq. (11):

$$E_H = E(\triangle I) = \sum_{i=0}^{L-1} p_i e^{(1-p_i)} \tag{11}$$

Then, we introduce $H_{local}(t)$ (Eq. (12)) and $H_{opt}(t)$ (Eq. (13)) denote the proportion of ants which were permitted to select routing and the probability of maintaining the optimal routing, respectively. In the begin of algorithm, according to the change in the entropy value, we can ensure smaller $H_{local}(t)$ value to search for the solution space as possible. In the later period, the larger $H_{local}(t)$ value can enhance Local optimization capability to avoid premature stagnation. Additionally, larger $H_{opt}(t)$ value contributes to find the optimal path as much as possible in the initial operation. In the later period, $H_{opt}(t)$ value decreased slowly in order to increase random operations and avoid premature phenomenon.

$$H_{local}(t) = \frac{S_{max} - S(t)}{\sqrt{\lambda} S_{max}} \tag{12}$$

$$H_{opt}(t) = 1 - \frac{S_{max} - S(t)}{\sqrt{\omega} S_{max}} \tag{13}$$

where $\lambda \in [0.8, 1.2]$, $\omega \in [3.8, 4.2]$ are constants.

The whole algorithm process is repeated until a stopping criterion is met. In the final cycle, the best ever solution found is submitted to 50 iterations of the ACA, for a final solution. For each isolated tour, the later stage is reduced to a TSP,and ILS can balance global exploration (distribution of consumers in tours, satisfaction of constraints) and local exploitation (minimizing tours). Gambardella et al.[26] have developed a multiple ant colony system for a more complex version of the VRPs. Their approach used two ant colonies to optimize a multiple objective function: the first colony minimizes the number of vehicles while the second colony minimizes the traveled distances. However, our work differs in the second stage where we apply iterated local search to minimize tours.

4 Experimental results

We now evaluate the performance of E2ACA in Java on a Pentium IV, 4GMB of RAM, 2.6 GHz processor. The algorithm was tested on two sets of benchmark problems. The 14 benchmark problems proposed by Christofides et al. [27] and the 20 large scale vehicle routing problems proposed by Golden et al. [28]. These include the best known solutions to each problem. The first benchmark problem contains between 51 and 200 nodes including the depot while second contains between 200 and 483 nodes including the depot. For each instance of the datasets, the number of customers is given by the first number on the instance name. The main difference between these sets of problems is their tightness (the ratio between demand and capacity) and the location of customers.

we used $n = 10$ artificial ants and set $\alpha = 1, q_0 = 0.9, \beta = 2$ and $\rho = 0.1$. For all problems maximum iteration times are $2 * n$. The quality of the generated solutions is given in terms of their relative deviation from the best known solutions (BKS), that is computed as $dev = (cost_{E2ACA} - cost_{BKS})/cost_{BKS} \times 100\%$, where $cost_{E2ACA}$ denotes the cost of the solution found by E2ACA, and $cost_{BKS}$ is the cost of the best known solution.

The results are reported in the Table 1 and Table 2, where the number of nodes of each instance is presented in the left column. The six columns of the right side present (i) the best solutions (BKS) that were known when our research started, (ii) the results of the best run of the E2ACA algorithm (best), (iii) the average results of the 50 runs of the E2ACA algorithm (average), (iv)the quality of the best run of the proposed algorithm (dev_{best}), (v)the average quality of the 50 runs of the algorithm (dev_{avg}) and (vi) the CPU time (in minute) of the best run

Table 1: Results of E2ACA in 14 benchmark Christofides instances

Instance	BKS	Best	Average	dev_{avg}	dev_{best}	CPU_{best}
vrpnc1	524.61 [29]	**524.61**	526.34	0.00	0.33	0.03
vrpnc2	835.26 [29]	837.26	839.38	0.24	0.49	0.12
vrpnc3	826.14 [29]	**826.14**	828.46	0.00	0.28	0.13
vrpnc4	1028.42 [29]	1031.34	1036.43	0.28	0.78	0.38
vrpnc5	1291.29 [29]	1316.45	1314.26	1.95	1.78	1.22
vrpnc6	555.43 [29]	**555.43**	557.23	0.00	0.32	0.03
vrpnc7	909.68 [29]	911.44	913.65	0.19	0.44	0.17
vrpnc8	865.94 [29]	**865.94**	867.25	0.00	0.15	0.53
vrpnc9	1162.55 [29]	1173.86	1176.86	0.97	1.23	0.86
vrpnc10	1395.85 [29]	1414.12	1418.61	1.31	1.63	1.81
vrpnc11	1042.11 [29]	1045.55	1049.19	0.33	0.68	0.16
vrpnc12	819.56 [29]	**819.56**	821.45	0.00	0.23	0.11
vrpnc13	1541.14 [29]	1547.55	1551.34	0.42	0.66	0.23
vrpnc14	866.37 [29]	867.98	871.32	0.19	0.57	0.19

Table 2: Results of E2ACA in 20 benchmark Golden instances

Instances	BKS	Best	Average	dev_{avg}	dev_{best}	CPU_{best}
Kelly01	5627.54 [2]	5627.43	5675.84	0.80	0.86	0.93
Kelly02	8444.5 [8]	8470.52	8477.15	0.31	0.39	1.41
Kelly03	11036.22 [30]	11109.72	11118.25	0.67	0.74	3.95
Kelly04	13624.52 [19]	13704.43	13712.61	0.59	0.65	5.73
Kelly05	6460.98 [31]	6472.26	6481.55	0.17	0.32	0.76
Kelly06	8412.88 [19]	8417.48	8425.76	0.05	0.15	0.74
Kelly07	10181.75 [32]	10221.97	10243.66	0.40	0.61	1.14
Kelly08	11643.9 [8]	11741.48	11763.21	0.84	1.02	3.91
Kelly09	583.39 [2]	**583.39**	583.39	0.00	0.00	0.44
Kelly10	741.56 [2]	745.78	747.26	0.57	0.77	1.67
Kelly11	918.45 [2]	922.41	926.75	0.43	0.90	1.94
Kelly12	1107.19 [2]	1117.09	1122.35	0.89	1.37	5.43
Kelly13	859.11 [2]	862.33	867.68	0.37	1.00	1.82
Kelly14	1081.31 [2]	1089.44	1097.43	0.75	1.49	1.23
Kelly15	1345.23 [2]	1349.33	1356.56	0.30	0.84	4.98
Kelly16	1622.69 [2]	1631.24	1636.78	0.53	0.87	6.18
Kelly17	707.79 [2]	**707.79**	708.36	0.00	0.08	1.04
Kelly18	997.52 [33]	1003.27	1008.32	0.58	1.08	1.69
Kelly19	1366.86 [2]	1379.98	1381.67	0.96	1.08	2.78
Kelly20	1820.09 [2]	1830.33	1837.78	0.56	0.97	3.15

of the proposed algorithm (CPU_{best}) are presented, respectively.

By exponential entropy, we observe that the entropy values are larger in the middle of the construction process of a solution than at the end or at the beginning of the process. A reason for this is that selection in the middle has a larger set of candidate customers, while selection at the beginning and at the end has similar entropy values. Also in this case, the entropy values are higher in later generations than in former generations. All entropy values are below 0.3 for the first 50 selections after generation 1000. This indicates an advantage of the combined evaluation method - it prevents the algorithm to converge too early.

In Table 2 the algorithm has found the best known solution in two of them (In Table 1 the number is five). For the rest instances, the averages obtained are close to the values of the best solutions, the quality is between 0.05% and 0.96% (In Table 1 the quality is between 0.19% and 1.95%) and the average quality of the best run for the twenty instances is 0.48% (In Table 1 the value is 0.42%).

In Tables, the computation time need (in minutes) for finding the best solution by E2ACA is presented. The problems are more complicated and the computational time is increased but is still less than 7 minutes for all instances. These results show the efficiency of our algorithm. In ten instances of both sets out of all 50 runs, the algorithm found the best known solution. The solutions found were very close to the best solutions even if the best solution was not found in all runs. It should be noted that we present a very fast and effective algorithm and the choice of the parameters was performed in such a way in order the algorithm to combine a fast convergence results.

5 Conclusion

Basic ant colony algorithm has some defects, such as slow convergence speed, easy to get stagnate, and low ability of full search. To overcome these problems, this paper proposes a hybrid algorithm which combines improved ant colony algorithm with one of the best local search algorithms, the iterated local search algorithm, for the merits of the two. By introducing the concept of exponential entropy, we use the entropy represents the uncertainty of the routing selection process to control the probability of routing selection and local random disturbance. Combined with the local optimization method, the second optimization for the solution is achieved. It overcomes the prematurity of the basic ant colony algorithm effectively making use of the pheromone updating strategy adaptively. Computational results show the efficiency of the proposed algorithm with respect to some standard benchmark problems.

References

[1] Tan, K.C., Chen, Y.H., & Lee, L.H. A Hybrid Multiobjective Evolutionary Algorithm for Solving Vehicle Routing Problem with Time Windows. Computational Optimization and Applications, **34**, 115-151 (2006).

[2] Mester, D., & Bräysy, O. Active-Guided Evolution Strategies for Large-Scale Capacitated Vehicle Routing Problems. Computers & Operations Research, **34**, 2964-2975 (2007).

[3] M. Dorigo, V. Maniezzo, & A. Colorni. The Ant System:Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics - Part B, **26**, 29-42 (1996).

[4] M. Dorigo, & L. M. Gambardella. Ant Colonies for the Traveling Salesman Problem, BioSystems, **43**, 73-81 (1997).

[5] Silvia Mazzeo & Irene Loiseau. An Ant Colony Algorithm for the Capacitated Vehicle Routing. Electronic Notes in Discrete Mathematics, **18**, 181-186 (2004).

[6] Kanyapat Watcharasitthiwat & Paramote Wardkein. Reliability optimization of topology communication network design using an improved ant colony optimization. Computers and Electrical Engineering, **35**, 730-747 (2009).

[7] O. Martin, S.W. Otto, & E.W. Felten. Large-Step Markov Chains for the Traveling Salesman Problem. Complex Systems, **5**, 299-326 (1991).

[8] Prins, C. A GRASP × evolutionary local search hybrid for the vehicle routing problem. In F. B. Pereira & J. Tavares (Eds.). Bio-inspired algorithms for the vehicle routing problem, SCI, Berlin, Heideberg: Springer, **161**, 35-53 (2008).

[9] Dantzig, G. B., & Ramser, J. H. The truck dispatching problem. Management Science, **6**, 80-91 (1959).

[10] Barbarosoglu, G., & Ozgur, D.A tabu search algorithm for the vehicle routing problem.Computers and Operations Research, **26**, 255-270 (1999).

[11] Cordeau, J. F., Gendreau, M., Laporte, G., Potvin, J. Y., & Semet, F. A guide to vehicle routing heuristics. Journal of the Operational Research Society, **53**, 512-522 (2002).

[12] Toth, P., & Vigo, D. The granular tabu search (and its application to the vehicle routing problem). INFORMS Journal on Computing, **15**, 333-348 (2003).

[13] Osman, I. H. Metastrategy simulated annealing and tabu search algorithms for combinatorial optimization problems. Annals of Operations Research, **41**, 421-451 (1993).

[14] Chen, P., Huang, H. K., & Dong, X. Y. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. Expert Systems with Applications, **37**, 1620-1627 (2010).

[15] Alabas-Uslu, C., & Dengiz, B. A self-adaptive local search algorithm for the classical vehicle routing problem. Expert Systems with Applications. doi: 10.1016/ j.eswa.2011.01.116.

[16] Yannis Marinakis. Multiple Phase Neighborhood Search-GRASP for the Capacitated Vehicle Routing Problem. Expert Systems with Applications. doi: 10.1016/j.eswa.2012.01.015.

[17] Baker, B. M., & Ayechew, M. A. A genetic algorithm for the vehicle routing problem.Computers and Operations Research, **30**, 787-800 (2003).

[18] Marinakis, Y., Migdalas, A., & Pardalos, P. M. A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm. Journal of Global Optimization, **38**, 555-580 (2007).

[19] Prins, C. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers and Operations Research, **31**, 1985-2002 (2004).

[20] Reimann, M., Doerner, K., & Hartl, R. F. D-ants: Savings based ants divide and conquer the vehicle routing problem.Computers and Operations Research, **31**, 563-591 (2004).

[21] Marinakis, Y., Marinaki, M., & Dounias, G. A hybrid particle swarm optimization algorithm for the vehicle routing problem. Engineering Applications of Artificial Intelligence, **23**, 463-472 (2010).

[22] R. Montemanni, L. Gambardella, A. Rizzoli, & A. Donati. A new algorithm for a dynamic vehicle routing problem based on ant colony system. In Second International Workshop on Freight Transportation and Logistics, (2003).

[23] M. Dorigo, V. Maniezzo & A. Colorni. Ant System: Optimization by a Colony of Cooperating Agents, IEEE Trans. Sys., Man, Cybernetics, **26**, (1996).

[24] B.Bullnheimer, R. F.Hartl, & Strauss, C. Applying the Ant System to the vehicle routing problem. Meta-heuristics:

Advances and trends in local search paradigms for optimization, 109-120 (1998).
[25] T. Stüutzle.Applying iterated local search to the permutation flow shop problem.Technical Report AIDA-98-04, FG Intellektik, TU Darmstadt, (1998).
[26] L.M. Gambardella, É. Taillard, and G. Agazzi. MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. In New ideas in optimization. D. Corne et al. editors, 63-76 (1999).
[27] Christofides, N., Mingozzi, A., & Toth, P. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, & C. Sandi (Eds.), Combinatorial optimization. Chichester: Wiley, (1979).
[28] Golden, B. L., Wasil, E. A., Kelly, J. P., & Chao, I. M. The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results. In T. G. Crainic & G. Laporte (Eds.), Fleet management and logistics. Boston: Kluwer Academic Publishers; 33-56 (1998).
[29] Taillard, E. D. Parallel iterative search methods for vehicle routing problems. Networks, **23**, 661-672 (1993).
[30] Reimann, M., Doerner, K., & Hartl, R. F. D-ants: Savings based ants divide and conquer the vehicle routing problem.Computers and Operations Research, **31**, 563-591 (2004).
[31] Tarantilis, C. D., & Kiranoudis, C. T. BoneRoute: An adaptive memory-based method for effective fleet management. Annals of Operations Research, **115**, 227-241 (2002).
[32] Pisinger, D., & Ropke, S. A general heuristic for vehicle routing problems. Computers and Operations Research, **34**, 2403-2435 (2007).
[33] Mester, D., & Bräysy, O. Active guided evolution strategies for the large scale vehicle routing problems with time windows. Computers and Operations Research, **32**, 1593-1614 (2005).

Chengming Qi is a faculty at the College of Automation, Beijing Union University, China. He received his Master of Engineering from the School of Information Engineering, China University of Geosciences (Beijing), 2004. His research interests include computational intelligence, logistics and supply chain management. He has published around 20 research papers. Corresponding author.

Ping Li is associate professor of College of Automation and director of institute of logistics, Beijing Union University, China. His research interests include computational intelligence, logistics and supply chain management. He has published around 10 research papers.