BAOZHEN YAO
E-mail: yaobaozhen@hotmail.com
PING HU
E-mail: huping_dl@163.com
MINGHENG ZHANG
E-mail: gloriazhang@163.com
School of Automotive Engineering,
Dalian University of Technology
Dalian, 116024, China
XIAOMEI TIAN
E-mail: txm055055@163.com
CTS International Logistics Corporation Limited
Dalian, 116001, China

Swarm Intelligence in Transportation Engineering
Original Scientific Paper
Accepted: Apr. 25, 2013
Approved: Feb. 12, 2014

IMPROVED ANT COLONY OPTIMIZATION FOR SEAFOOD PRODUCT DELIVERY ROUTING PROBLEM

ABSTRACT

This paper deals with a real-life vehicle delivery routing problem, which is a seafood product delivery routing problem. Considering the features of the seafood product delivery routing problem, this paper formulated this problem as a multi-depot open vehicle routing problem. Since the multi-depot open vehicle routing problem is a very complex problem, a method is used to reduce the complexity of the problem by changing the multi-depot open vehicle routing problem into an open vehicle routing problem with a dummy central depot in this paper. Then, ant colony optimization is used to solve the problem. To improve the performance of the algorithm, crossover operation and some adaptive strategies are used. Finally, the computational results for the benchmark problems of the multi-depot vehicle routing problem indicate that the proposed ant colony optimization is an effective method to solve the multi-depot vehicle routing problem. Furthermore, the computation results of the seafood product delivery problem from Dalian, China also suggest that the proposed ant colony optimization is feasible to solve the seafood product delivery routing problem.

KEY WORDS

Seafood Product Delivery Routing Problem, Multi-Depot Open Vehicle Routing Problem, Ant Colony Optimization, Adaptive Strategy, Crossover Operation

1. INTRODUCTION

Seafood processing, distribution and selling have been the main roles for a seafood company. Some literature concerned with these problems provides the awareness of the necessity of solving the problem concerned with seafood [1, 2]. The “Seafood”, from the point of its definition, should include finfish, shellfish and crustaceans in fresh, frozen and processed product forms. In this paper the seafood in processed product forms is considered, noted as “seafood product”. With the increasing demands of seafood product, the seafood product delivery cost constitutes a significant part of the operational costs of a seafood product company.

For a seafood company to ensure the quality of seafood product, the locations of the factories are mostly near the ports. In the factories, seafood will be processed according to the packaging standards. Then, there are many seafood product factories whose seafood products should be delivered from these factories to a set of customers (for example, hypermarkets). The management of the seafood company estimated that the fixed cost of the vehicles involved in the delivery process of seafood product is high enough. Therefore, the company decided to hire vehicles to complete the delivery tasks which cover only the distances from seafood product factories to the customers. Since the delivery cost paid by the seafood company is proportional to the total travel cost, the paper attempts to optimize the delivery routes for vehicles that minimize the total travelling and vehicle operating costs of the delivery process. The fact that the hired vehicles and the depot of the seafood product factories is not simple, led the authors to formulate this real-life problem as a combination of an open VRP(OVRP) [3, 4] and a multi-depot vehicle routing problem (MDVRP) [5, 6]. However, these vehicle delivery routing problems have the following common characteristics: all the vehicles have the same capacity; the number of vehicles is unrestricted; each customer is visited once and only once by one of the vehicles and its demand must be completely fulfilled; the total demand of all the customers on a route must not exceed the capacity of the vehicle; each vehicle makes one trip only; the objective is to find the set of routes, and minimize the total distance travelled by the vehicles.

From the view of the model, both the OVRP and MDVRP do not completely fit our real-life problem. However, the concept of OVRP is involved in the formulation of the problem since it allows the hired vehicles to be assigned to routes in which they do not have to return to the seafood product factory after serving the customers, providing the company with the set of routes that minimizes the total distance travelled. Furthermore, the delivery process starts at more than one depot. Therefore, it will turn out that the model of the multi-depot seafood product delivery problem belongs to the open MDVRP (MDOVRP) based on the idea of OVRP and MDVRP.

Since both OVRP and MDVRP are considered as NP-hard problems, it is difficult to solve them by traditional methods. Recently, many studies have proved that heuristic techniques are suitable for solving this kind of complicated problems [6, 7-15]. Brandao [4] presented a tabu search algorithm to solve the open vehicle routing problem, which uses an innovative concept that consists of integrating the lower bounds of a problem. Repoussis *et al.* [16] introduced a hybrid evolution strategy to solve the open vehicle routing problem and the results on well-known benchmark data sets suggested the effectiveness of the algorithm for this problem. Salari *et al.* [3] presented a heuristic improvement procedure based on integer linear programming techniques for solving the open vehicle routing problem. Mirabi *et al.* [17] presented a hybrid heuristics for solving the multi-depot vehicle routing problem. In the three hybrid heuristics, a constructive heuristic search and improvement techniques have been used to improve the performance of the algorithm in each hybrid heuristics. Ho *et al.* [18] presented a hybrid genetic algorithm to solve the multi-depot routing problem. In the hybrid genetic algorithm, the initial solutions are generated randomly in one hybrid genetic algorithm and the initialization procedure is performed in other hybrid genetic algorithms. Yu *et al.* [6] attempted to solve the multi-depot vehicle routing problem with an improved ant colony optimization. In the algorithm, a coarse-grain parallel strategy, ant weight strategy and mutation operation have been used to improve the performance of ant colony optimization.

Ant colony optimization (ACO), an artificial intelligence procedure inspired by food-seeking behaviours of ant colonies in nature, was first proposed by Dorigo *et al.* [19]. It has been successfully applied to solve these kind of complicated optimization problems, e.g. travelling salesman [19] quadratic assignment [20], job-shop scheduling [21], telecommunication routing [22], vehicle delivery routing problem [23], etc. Therefore, this paper also adopts ACO to solve the seafood product delivery routing problem.

The remainder of this paper is organized as follows. Section 2 describes the seafood product delivery routing problem as a multi-depot open delivery routing problem. Section 3 presents ACO and introduces some improving strategies to enhance the performance of ACO. Some computational results are discussed in Section 4 and finally, the conclusions are provided in Section 5.

2. PROBLEM FORMULATION

2.1 Seafood product delivery Routing Problem

The seafood product delivery Routing Problem can be described as follows: There are m seafood product depots, each of which owns K_i vehicles with the same capacity. All the vehicles are responsible for the service of n customers, the demand of customer i are d_i $(i = 1, \ldots, n)$, and $d_i < q$ (q is the limit of vehicle load). The distance between two customers or between customers and the delivery points is described as the vertex set C which is partitioned into two subsets: $C_d = \{c_1, \ldots, c_H\}$ is the set of automobile part depots and $C_c = \{c_{H+1}, \ldots, c_{H+N}\}$ is the set of customers, respectively. The distance matrix is a real symmetric one satisfying the triangle inequality principle, that is, $c_{ik} \le c_{ij} + c_{jk}$. Each customer can be served by only one vehicle from any depot. Each vehicle can serve several customers whose demands must not exceed the transportation capacity of the vehicle. The objective of the seafood product delivery routing problem is to minimize the number of vehicles and, for a given number of vehicles, to minimize the total distance travelled by the vehicles. In the seafood product delivery routing problem, each customer can be served by only one vehicle from any depot. Each vehicle can serve several customers whose demands must not exceed the transportation capacity of this vehicle. And each route is a sequence of customers which should start from one seafood product depot and finish at one of the customers. In this paper, the number of vehicles is given and the objective of the seafood product delivery routing problem is to minimize the total delivery distance in serving all customers while it meets the following constraints. Thus, the model of the seafood product delivery routing problem is described as follows:

$$\min \sum\sum\sum\sum C_{ij} x_{ij}^{mk} \tag{1}$$

subject to:

$$\sum_{j=1}^{N+M}\sum_{m=1}^{M}\sum_{k=1}^{K_m} x_{ij}^{mk} = 1 \quad \text{for } i \in \{1,\ldots,N\} \tag{a}$$

$$\sum_{i=1}^{N} q_i \sum_{j=1}^{N+M} x_{ij}^{mk} \le q_{mk} \quad \text{for } m \in \{N+1, N+2, \ldots, N+M\}; \quad k \in \{1,2,\ldots,k_m\} \tag{b}$$

$$\sum_{j=1}^{N+M}\sum_{m=1}^{M}\sum_{k=1}^{K_m} x_{ij}^{mk} = 1 \quad \text{for } j \in \{1,\ldots,N\} \tag{c}$$

$$\sum_{j=N+1}^{N+M} x_{ji}^{mk} = \sum_{j=N+1}^{N+M} x_{ij}^{mk} = 0 \quad \text{for } i = m \in \{N+1, N+2, \ldots, N+M\}; \quad k \in \{1,2,\ldots,k_m\} \tag{d}$$

where

$$x_{ij}^{mk} = \begin{cases} 1 & \text{the link from customer } i \text{ to } j \text{ is visited by vehicle } k \text{ from depot } m \\ 0 & \text{otherwise} \end{cases}$$

Constraint (a) service constraints: to assume that every customer node can be visited exactly once by one vehicle. Constraint (b) the capacity constraint: to assume the load of a vehicle cannot exceed its capacity. Constraint (c) the maximum number of vehicles constraint: to assure the number of vehicles used is less than the maximum. Constraint (d) ensures a vehicle serves a customer and leaves it.

The seafood product delivery routing problem can be described as in *Figure 1*.

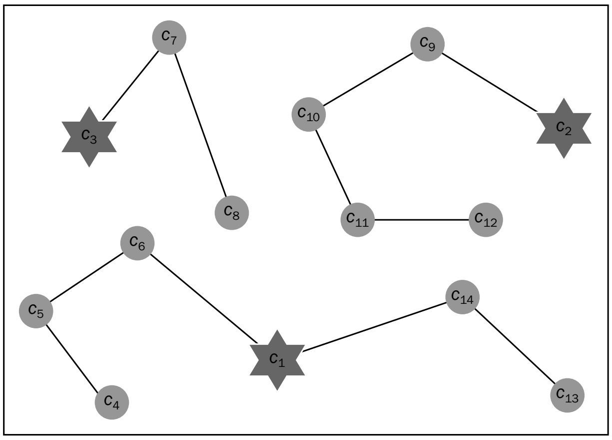

Figure 1 - An example of the seafood product delivery routing problem where circles represent customers, and stars represent depots.

2.2 Seafood product delivery problem with a dummy depot

Since the delivery routes consist of a combination of whichever depot and customers, the seafood product delivery problem is NP-hard and very difficult to solve even for relatively small size instances. It is necessary to reduce the complexity of the problem. In this paper, a method [6] is adopted to simplify the seafood delivery problem. First, a dummy depot c_0, whose distances to H actual depots and N customers are assumed as 0 and ∞, respectively, is added. Each vehicle starts its route at the added dummy depot through one actual depot to a sequence of customers without exceeding the capacity constraints of each vehicle. Thus, the seafood delivery problem can be changed into a seafood delivery problem with a dummy depot, which is similar to an OVRP with the dummy depot as the origin. *Figure 2* is an example of the seafood delivery problem with a dummy depot transferred from the same problem of *Figure 1* by using a dummy depot.

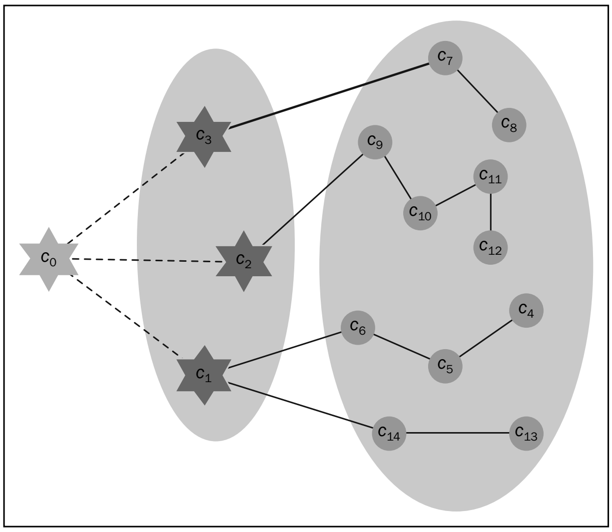

Figure 2 - An example of the seafood delivery problem with a dummy depot

3. IMPROVED ACO FOR THE SEAFOOD PRODUCTION DELIVERY PROBLEM WITH A DUMMY DEPOT

The solution of the seafood delivery problem with a dummy depot is to find a set of minimum cost routes in order to facilitate the delivery from the dummy depot through m actual depot to a number of customer locations. This is very similar to food-seeking behaviour of ant colonies in nature. If we take the dummy depot as the *nest*, the actual depots as the entries of the nest and the customers as the *food*, the problem can be described as a process by which the ant colony searches for *food* starting from the *nest* through an entry. The general description of ACO can be seen in the literature [12, 23]. The specific steps of the improved ACO are as follows.

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. Initially, each ant starts at the central depot and the set of customers included in its route is empty. The ant selects the next customer to visit from the list of feasible locations and the storage capacity of the vehicle is updated before another customer is selected. The ant returns to the central depot when the capacity constraint of the vehicle is met or when all customers are visited. The total cost is computed as the objective function value for the complete route of the artificial ant. The ACO algorithm constructs a complete route for the first ant prior to the second ant starting its route. This continues until the last ant constructs a feasible route. In the iterative phase, the customers or partial routes are combined by sequentially choosing feasible entries from the actual depots and the customers. A combination is infeasible if it violates the capacity constraints or the customer has already been visited.

The decision making about combining customers is based on a probabilistic rule taking into account both the visibility and the pheromone information. The pheromone can tell us how good the combination of these two customers *i* and *j* was in previous iterations. The visibility is constructed based on the well-known savings Algorithm due to [24] and it contains the savings of combining two customers *i* and *j* on one route as opposed to serving them on two different routes. Thus, to select the next customer *j* for the k_{th} ant at the i_{th} node, the ant uses the following probabilistic formula.

$$p_{ij}(k)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}\times\eta_{ij}^{\beta}}{\sum_{h\notin tabu_k}\tau_{ih}^{\alpha}\times\eta_{ih}^{\beta}} & j\notin tabu_k\\ 0 & \text{otherwise}\end{cases}\tag{2}$$

where $p_{ij}(k)$ = the probability of choosing to combine customers *i* and *j* on 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; *tabu* = the set of infeasible nodes for the k_{th} ant.

3.2 Adaptive strategy for parameters α and β

It can be attained from Formula (2) that two parameters α and β have important roles in ACO. Parameter α is a factor of the information heuristic to reflect the effect of pheromone. Parameter β is a factor of the expectation heuristic to express the relative importance of the visibility. Thus, the two parameters can greatly affect the randomness and determination of the algorithm. In general, before launching the algorithm the value of the two parameters generally are determined as a constant according to the user's experience, or by an external tuning method. The main drawback of the determination for the two parameters is that the values are constants during the searching process of the algorithm. However, the functions of the two parameters have in fact different effects during the searching process of the algorithm.

Parameter α determines the extent of randomization during the evolution process. The larger the effect of randomization the larger is the diversity while small value of parameter α can ensure a reasonable convergence velocity. Similarly, parameter β determines the extent of determination during the evolution process. The larger the effect of determination the larger is the convergence velocity while a small value of parameter β can ensure the diversity. Therefore, in the early stage of the searching process, parameter α with a small value and parameter β with a lager value can ensure a reasonable convergence velocity while parameter α with a larger value and parameter β with a small value can help the algorithm to search a wider space. More diversity leads to the algorithm escaping from the local optima during later stages. Therefore, a self-adaptive strategy for parameters α and β is introduced to improve the searching ability of the algorithm.

$$\alpha=\left\lfloor\frac{t\times 3}{T\max}\right\rfloor+1\tag{3}$$

$$\beta=3-\left\lfloor\frac{t\times 2}{T\max}\right\rfloor\tag{4}$$

where, *t* is the current iteration and $T\max$ is the maximum number of iterations. Equations (3) and (4) provide a self-adaptive, non-linear model for optimizing the value of parameters α and β which is used to balance the global and local searching ability of the algorithm at a reasonable convergence velocity.

3.3 The 2-opt exchange

In the searching process of the algorithm, 2-opt is a simple local search algorithm [25] which is also applied to ensure local optimality. The main idea of 2-opt is to improve the routing by deleting two edges and by replacing them with the only possible couple of new edges in the route constraint. In the 2-opt exchange, all the possible pairwise exchanges of customer locations visited by individual vehicles are examined to see if an overall improvement in the objective function can be attained. If all pairs of the edges were considered and no further solution is found, a pareto local optimum solution is reached. The method has been used in several ACOs [23, 26-29] for VRP.

3.4 Crossover operation

Crossover is a genetic operator which exchanges genetic information between two parent chromosomes to form two new children at a predefined probability p_c. The operators can help the algorithm to search a larger space. The idea of crossover operation is to randomly change two tours and produce two new solutions. In this paper, the crossover operator is designed to conduct customer exchanges in a random fashion. Firstly, two routes in the solution are selected

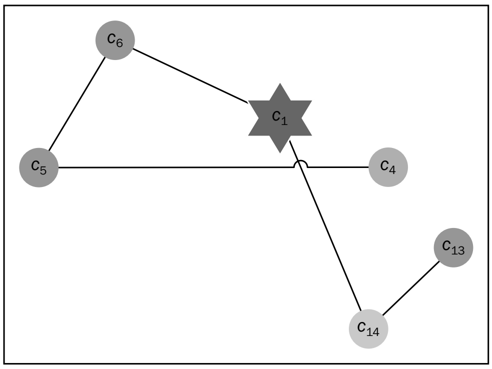

Figure 3 - Selecting the points of the crossover

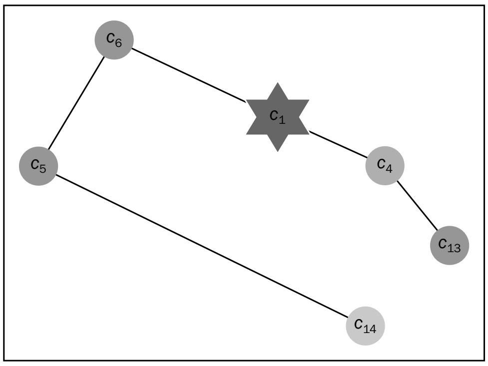

Figure 4 - Exchanging the two points of the crossover

randomly and one customer in each selected route is selected randomly. Then, the two selected customers are exchanged and a new solution is created. *Figure 3* shows the selection operation for the crossover operation and *Figure 4* shows the exchange process for the crossover operation.

To achieve a better performance, the crossover rate p_c setting should be done while solving the problem. It is important to control the amount of time invested for the crossover operation. While the crossover rate p_c with a large value will increase the computation time, the crossover probability p_c with a large value will decrease the efficiency of the algorithm. Therefore, a balance between a reasonable computation time and efficiency is necessary. Usually, the diversity of the solution is large at the beginning of a run and decreases with time. We adopt the crossover rate during a run to promote fast convergence to good solutions during the first generations and to introduce more diversity for escaping from local optima during later stages. The crossover rate p_c at generation t is:

$$p_c(t) = p_c^{\min} + (p_c^{\max} - p_c^{\min})^{1 - t/T_{\max}} \tag{5}$$

where $p_c^{\min}$ and $p_c^{\max}$ denote the lower and upper crossover rates for the beginning and ending, respectively. $T_{\max}$ and t are the maximum number of generations and the current generation, respectively.

According to preliminary tests, the lower crossover rate is set to $p_m^{\min} = 1/n_c$ and n_c is the number of the customers. The upper crossover rate is set to $p_m^{\max} = 1/n_v$ and n_v is the amounts of the routes in the solution.

3.5 Update of Pheromone information

In order to improve future solutions, the pheromone trails of the ants must be updated to reflect the ant's performance and the quality of the solutions found. This updating is the key element to the adaptive learning technique of ACO and helps ensure the improvement of subsequent solutions. First, Pheromone updating is conducted by reducing the amount of Pheromone on all visited arcs in order to simulate the natural evaporation of Pheromone and to ensure that no one path becomes too dominant. This is done with the following Pheromone updating equation,

$$\tau_{ij} = \rho \times \tau_{ij} + \sum_k \Delta\tau_{ij}^k \quad \rho \in (0,1) \tag{6}$$

where Δt_{ij}^k = the Pheromone increments of the kth route in the best solution on edge (i,j); ρ = the parameter that controls the speed of evaporation; k = the No. of the route.

From Formula (8) it can also be found that parameter ρ controls the quality of the Pheromone of the routes by a predefined value between 0 and 1. In the early stage, a larger value of parameter ρ is sufficient to simulate the natural evaporation. However, in the later stage, it is necessary to decrease the value of parameter ρ to improve the searching ability when the optimum routes have not improved after consecutive iterations. If the value of parameter ρ is too small, it will greatly affect the convergence velocity. Thus, an adaptive strategy for parameter ρ is used in this paper.

$$\rho^{(t)} = 1 - (1 - \rho_{\min})^{\frac{t}{T_{\max}}} \tag{7}$$

where $\rho_{\min}$ is the lower value for parameter ρ.

There are three popular Pheromone increment updating strategies: a) Ant-density; (b) Ant-quantity; and (c) Ant-cycle. Some of them omit global information, while others omit local information. This may result in false directive information and a large amount of invalid searches. Therefore, we have adopted a new Pheromone increment updating strategy: Ant-weight, which takes into account both global and local information. Specifically, the Pheromone increment update strategy [23] is written as:

$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{f^k} \times \dfrac{f^k - f_{ij}^k}{(n-2)f^k} & \text{when the } k_{th} \text{ ant visits edge } (i,j) \text{ in the } t\text{-th cycle} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where Q = a constant; f^k = the object value of the solution; f_{ij}^k = the object value of the k_{th} route on edge (i,j) in the solution; n = the number of routes and $n > 2$.

This increment update strategy consists of two components: the first one is global Pheromone increment Q/f^k that is related to the quality of the k_{th} optimization route; the second one is local Pheromone increment

$$\frac{f^k - f_{ij}^k}{(n-2)f^k}$$

that is related to the contribution of edge (i,j) to the optimization route. In this way, 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. At the same time, the effectiveness of a large-scale search can be ensured to facilitate the algorithm to find the overall optimal path. Additionally, to prevent local optimization and to enlarge the probability of gaining a higher-quality solution, an upper limit and a lower limit $[\tau_{\min}, \tau_{\max}]$ are introduced to the Pheromone on each link [30].

$$\tau_{\min} = Q/\sum_i 2d_{0i}, \tag{9}$$

$$\tau_{\max} = Q/\sum_i d_{0i} \tag{10}$$

where d_{0i} = the distance from the central depot to the i_{th} customer.

4. NUMERICAL ANALYSIS

This paper attempts to use an improved ACO to solve a real-life delivery route problem which can be considered as MDOVRP. Since there is little mention in literatures of heuristics for solving MDOVRP, it is difficult for us to compare the adaptive algorithm with the previous methods. For comparison, some well-known benchmark problems of MDVRP were applied to examine the performance of the improved ACO. Then, the improved ACO was used to solve the seafood product delivery routing problem. The following will describe the two examples respectively.

4.1 Classical MDVRP

To examine the performance of the improved ACO in this paper, a set of instances derived from the MDVRP instances from literature [31-33] were used. These instances and the best known solutions are available at http://neo.lcc.uma.es/radi-aeb/WebVRP//index.html?/Problem Instances/MDVRPInstances.html. The main characteristics of these test problems are summarized in *Table 1*. The improved ACOs were coded in Visual studio. NET 2003: C++ and executed on a PC equipped with 3.25 GB of RAM and a Pentium processor running at 2.93 GHz. A simulation was performed to determine the parameters in the improved ACO. The parameter Q is set at 1,000 and the maximum number of generations is set at 1,000. In order to examine the performance of the algorithm, the results of the improved ACO algorithm were compared with FIND algorithm [5, 34, 6]. *Table 2* shows the results from these algorithms for solving MDVRP.

Table 1 - Information of the test problems

No.	H	n	Q	No.	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				

From *Table 2* it can be found that the efficiency and effectiveness of the improved ACO compared these well-known algorithms. Most of the results produced from the improved ACO have been close to the best solutions among these algorithms. It suggests that the improved ACO is suitable for MDVRP. Furthermore, some solutions are better than the given best solutions; thus, the improved ACO is fairly effective to solve MDVRP. To further test the performance of the adaptive strategy for ACO in this paper, a standard ACO and the improved ACO are used to solve the MDVRP simultaneously. The results are shown in *Table 3*.

It can be observed that the computation times and the solutions from the two algorithms are almost equal to solve some simple problems such as problems 1, 2 and 3. It is also found that when the problems are more complex, the computation times of the adaptive algorithm are more than the ones of the standard ACO. However, the solutions from the improved ACO are obviously better than the ones from the standard ACO. It can be explained that the adaptive strategy can explore larger solution space and prevent the algorithm from being trapped in the local optimum. However, the adaptive strategy also needs more time to enlarge the search space in the later iteration. Furthermore, we can find that the computation times are almost equal when solving larger problems such as problems 21, 22 and 23. It is due to the fact that a dummy depot is added to greatly improve the searching velocity when solving larger problems.

4.2 Seafood product delivery routing problem

The performance of the proposed algorithm is examined on the benchmark problems of MDVRP

Table 2 - Computational results of some well-known published results and the improved ACO

No.	Best-known solutions	FIND [5]	CGL [34]	PIACO [6]	Improved ACO
1	576.86	576.86	576.86	576.86	576.86
2	473.53	473.53	473.87	473.53	484.28
3	641.18	641.18	645.15	641.18	641.18
4	1,001.49	1,003.86	1,006.66	1,001.49	1,006.66
5	750.26	750.26	753.40	750.26	750.26
6	876.50	876.50	877.84	876.50	878.34
7	885.69	892.58	891.95	885.69	891.95
8	4,437.58	4,485.08	4,482.44	4,482.38	4,482.44
9	3,900.13	3,937.81	3,920.85	3,912.23	3,920.85
10	3,663.00	3,669.38	3,714.65	3,663.00	3,669.38
11	3,554.08	3,648.94	3,580.84	3,554.08	3,554.08
12	1,318.95	1,318.95	1,318.95	1,318.95	1,318.95
13	1,318.95	1,318.95	1,318.95	1,318.95	1,318.95
14	1,360.12	1,365.68	1,360.12	1,365.68	1,360.12
15	2,505.29	2,551.45	2,534.13	2,551.45	2,551.45
16	2,572.23	2,572.23	2,572.23	2,572.23	2,572.23
17	2,708.99	2,731.37	2,720.23	2,708.99	2,720.23
18	3,702.75	3,781.03	3,710.49	3,781.03	3,710.49
19	3,827.06	3,827.06	3,827.06	3,827.06	3,827.06
20	4,058.00	4,097.06	4,058.07	4,097.06	4,097.06
21	5,474.74	5,656.46	5,535.99	5,474.74	5,495.54
22	5,702.06	5,718.00	5,716.01	5,772.23	5,718.00
23	6,095.36	6,145.58	6,139.73	6,125.58	6,183.13

which is in complete Euclidean graphs with straight-line distance. However, in the real-life seafood product delivery routing problems, the distance between two points is based on the length of the routes. *Figure 4* shows the location information of the real-life seafood product delivery routing problems in Dalian. There are several seafood product depots and some hypermarkets in Dalian city which can be seen in *Figure 5*. The problem is how these seafood product depots send the seafood product to the hypermarkets with the least delivery cost (the least length of delivery routes).

Figure 5 - The location information of the seafood product delivery routing problem

As shown in *Figure 5* the rhombi stand for four seafood product depots in Dalian, and the dots represent the geographical positions of several hypermarkets in Dalian. The vehicle capacity is 200 and the specific information of consumers and distribution centres is shown in Tables 4 and 5, respectively.

Figure 6 shows the computing results of the proposed ACO algorithm while it continues calculating the seafood product delivery routing problem ten times. It can be found that the results are stable and the difference between the optimum and the worst plan is less than 3%. Meanwhile, the computing time, from 280 to 460 seconds, is very short for the larger complicated problems. Therefore, the algorithm has an excellent convergence performance to solving the seafood product delivery routing problem. The length of the routes is 66,375 KM and the optimized routes of the seafood product delivery routing problem can be shown in *Figure 7*.

Table 3 - Comparison results between ACO and the improved ACO

No.	H	n	ACO		Improved ACO	
			length	time	length	time
1	4	50	576.86	60	576.86	58
2	4	50	484.28	62	484.28	60
3	5	75	645.16	120	641.18	117
4	2	100	1,020.52	134	1,006.66	136
5	2	100	750.26	166	750.26	171
6	3	100	878.34	156	878.34	165
7	4	100	898.80	171	891.95	178
8	2	249	4,508.14	863	4,482.44	876
9	3	249	4,083.44	855	3,920.85	867
10	4	249	3,747.62	953	3,669.38	959
11	5	249	3,599.93	879	3,554.08	884
12	2	80	1,327.00	102	1,318.95	103
13	2	80	1,318.95	93	1,318.95	92
14	2	80	1,375.22	90	1,360.12	91
15	4	160	2,588.22	330	2,551.45	338
16	4	160	2,604.9	447	2,572.23	452
17	4	160	2,776.99	458	2,720.23	463
18	6	240	3,907.88	774	3,710.49	776
19	6	240	3,863.03	805	3,827.06	810
20	6	240	4,231.28	840	4,097.06	845
21	9	360	5,579.86	3,111	5,495.54	3,108
22	9	360	5,897.64	2,446	5,718.00	2,440
23	9	360	6,341.61	1,948	6,183.13	1,945
Avg	-	-	2,739.39	689.70	2,683.89	691.65

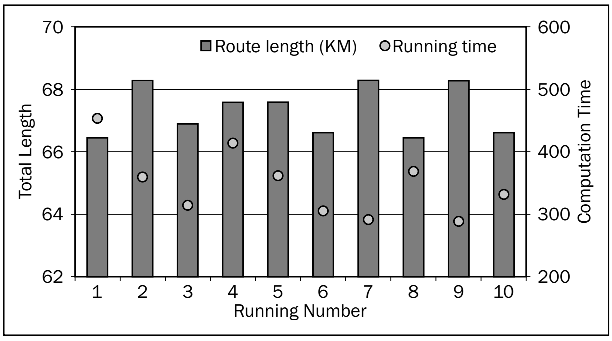

Figure 6 - Computing results of the improved ACO after running ten times

Figure 7 - Distribution routes of the seafood product delivery routing problem based on the improved ACO

5. CONCLUSION

We have considered the seafood product delivery routing problem in Dalian city as MDOVRP. In this problem, the seafood product depots expand and integrate with each other to keep their competitiveness in the market by reducing the delivery cost. The presence of multi-depot adds considerable difficulties to the standard VRP. A method is used to reduce the complexity of the problem by changing the MDOVRP into an OVRP with a dummy central depot in this paper. Then an improved ACO is presented to solve this seafood product delivery routing problem. In the algorithm, adaptive strategies for the selection of parameters α, β, ρ and the crossover rate are proposed to improve the performance of ACO. The algorithm was tested on the benchmark problems of MDVRP and the real-life sea-

Table 4 - The information of the hypermarkets in Dalian

Depots	1	2	3	4	5	6	7	8
Longitude	38.9072	38.9184	38.9458	38.876	38.9696	38.9293	38.9198	38.8861
Latitude	121.608	121.587	121.57	121.548	121.599	121.6	121.656	121.633
Demand	15	50	50	18	50	30	30	20
Depots	9	10	11	12	13	14	15	16
Longitude	38.8942	38.9663	38.9742	38.9577	38.9576	38.925	38.9723	38.8846
Latitude	121.586	121.517	121.557	121.6	121.564	121.545	121.613	121.651
Demand	20	50	20	80	60	60	36	25
Depots	17	18	19	20	21	22	23	24
Longitude	38.9184	38.9229	38.8919	38.9198	38.882	38.8848	38.9085	38.885
Latitude	121.674	121.641	121.676	121.625	121.529	121.561	121.587	121.584
Demand	20	20	40	20	15	20	40	20
Depots	25	26	27	28	29	30	31	32
Longitude	38.9186	38.9225	38.9008	38.8987	38.9155	38.9055	38.9263	38.9264
Latitude	121.601	121.601	121.599	121.644	121.636	121.666	121.658	121.638
Demand	50	20	18	30	25	20	20	18
Depots	33	34	35	36	37	38	39	40
Longitude	38.9276	38.9089	38.8646	38.8872	38.8837	38.8766	38.913	38.9687
Latitude	121.619	121.63	121.626	121.7	121.546	121.564	121.661	121.59
Demand	18	18	20	20	18	15	40	60
Depots	41	42	43					
Longitude	38.9729	38.9404	38.8936					
Latitude	121.526	121.577	121.623					
Demand	30	70	20					

Table 5 - The information of seafood product manufacture depots

Depots	A	B	C
Longitude	121.627	121.635	121.609
Latitude	38.9597	38.9345	38.9394

food product delivery routing problem. The algorithm is proved effective to solve these problems and at reasonably fast running times. Therefore, the proposed improved ACO can be considered as an effective method for the seafood product delivery routing problem.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China 51208079 and 11272075, the National Science Foundation for Postdoctoral Scientists of China 2013M530924.

摘要

基于改进蚁群算法的海产品配送路径问题

本文所研究的海产品配送路径问题属于车辆配送路径问题的一种。结合海产品配送的特点，本文将海产品配送路径问题设定为多中心开放式车辆路径问题。为降低多中心开放式车辆路径问题的复杂性，本文通过引入虚拟中心，从而将多中心开放式车辆路径问题转化为单中心开放式车辆路径问题。由于此问题比较复杂，因此，本文采用蚁群算法来优化该问题。此外，为了提高蚁群算法的求解性能，我们引入交叉操作和自适应策略。最后，通过若干经典多中心车辆路径实例对该算法进行了检验，结果表明本文提出的改进蚁群算法是一种求解多中心车辆路径问题的有力工具，同时通过实际问题的求解也证明了本文所采用的改进的蚁群算法来求解海产品配送路径问题是有效的。

关键词

海产品配送路径问题；多中心开放式车辆路径问题；蚁群算法；自适应策略；交叉操作

REFERENCES

[1] **Jaffry, S., Pickering, H., Ghulam, Y., Whitmarsh, D., Wattage, P.**: (2004); *Consumer choices for quality and sustainability labelled seafood products in the UK*. Food Policy 29(3):215–228

[2] **Guillotreau, P.**, **Peridy, N.**: (2000); *Trade barriers and European imports of seafood products: a quantitative assessment*. Marine Policy, 24(5):431-437

[3] **Salari, M.**, **Toth, P.**, **Tramontani, A.**: (2010); *An ILP improvement procedure for the Open Vehicle Routing Problem*. Computers & Operations Research, 37(12):2106-2120.

[4] **Brandao, J.**: (2004); *A tabu search algorithm for the open vehicle routing problem*. European Journal of Operational Research, 157(3):552-564

[5] **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

[6] **Yu, B.**, **Yang, Z.Z.**, **Xie, J.X.**: (2011); *A parallel improved ant colony optimization for multi-depot vehicle routing problem*. Journal of the Operational Research Society, 62:183-188

[7] **Chen, G.**, **Govindan, K.**, and **Yang, Z.Z.**: (2013); *Managing truck arrivals with time windows to alleviate gate congestion at container terminals*, International Journal of Production Economics, 141(1):179-188

[8] **Chen, G.**, **Yang, Z.Z.**: (2010); *Optimizing Time Windows for Managing Arrivals of Export Container in Chinese Container Terminals*, Maritime Economics & Logistics, 12(1):111-126

[9] **Yao, B.Z.**, **Hu, P.**, **Lu, X.H.**, **Gao, J.J. Zhang, M.H.**: (2013); *Transit network design based on travel time reliability*. Transportation Research Part C, DOI:10.1016/j.trc.2013.12.005(in press).

[10] **Yao B.Z.**, **Hu, P.**, **Zhang, M.H.**, **Wang, S.**: (2013); *Artificial Bee Colony Algorithm with Scanning Strategy for Periodic Vehicle Routing Problem*. SIMULATION: Transactions of the Society for Modeling and Simulation International. 89(6):762-770

[11] **Yao, B.Z.**, **Yang, C.Y.**, **Yao, J.B.**, **Sun, J.**: (2010); *Tunnel Surrounding Rock Displacement Prediction Using Support Vector Machine*. International Journal of Computational Intelligence Systems, 3(6): 843-852

[12] **Yu, B.**, **Yang, Z.Z.**: (2011); *An ant colony optimization model: The period vehicle routing problem with time windows*. Transportation Research Part E, 47(2):166-181

[13] **Yu, B.**, **Lam, W.H.K.**, **Lam, T.M.**: (2011); *Bus Arrival Time Prediction at Bus Stop with Multiple Routes*. Transportation Research Part C, 19(6):1157-1170

[14] **Yu, B.**, **Yang, Z.Z.**, **LI, S.**: (2012); *Real-Time Partway Deadheading Strategy Based on Transit Service Reliability Assessment*. Transportation Research Part A, 46(8):1265-1279

[15] **Yue, M.**, **Zhang, Y.S.**, **Tang, F.Y.**: (2013); *Path following control of a two-wheeled surveillance vehicle based on sliding mode technology*. Transaction of the Institute of Measurement and Control, 35(2): 212-218

[16] **Repoussisa, P.P.**, **Tarantilisa, C.D.**, **Bräysyb, O.**, **Ioannoua, G.**: (2010); *A hybrid evolution strategy for the open vehicle routing problem*. Computers & Operations Research, 37(3):443-455

[17] **Mirabi, M.**, **Fatemi Ghomib, S.M.T.**, **Jolaic, F.**: (2010); *Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem*. Robotics and Computer-Integrated Manufacturing, 26(6):564-569

[18] **Ho, W.**, **Ho, G.T.S.**, **Ji, P.**, **Lau, H.C.W.**: (2008); *A hybrid genetic algorithm for the multi-depot vehicle routing problem*. Engineering Applications of Artificial Intelligence, 21(4):548-557

[19] **Dorigo, M.**, **Maniezzo, V.**, **Colorni, A.**: (1996); *The Ant System: Optimization by a Colony of Cooperating Agents*, IEEE Transactions on Systems, Mans, and Cybernetics 1 (26), 29-41

[20] **Gambardella, L.**, **Taillard, E.**, **Dorigo, M.**: (1997); *Ant Colonies for the QAP*, Technical Report 97-4, IDSIA, Lugano, Switzerland

[21] **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

[22] **Schoonderwoerd, R.**, **Holland, O.**, **Bruten, J.**, **Rothkrantz, L.**: (1997); *Ant-Based Load Balancing in Telecommunications Networks*, Adaptive Behavior 5 (2), 169-207

[23] **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

[24] **Clarke, G.**, **Wright, J.W.**: (1964); *Scheduling of Vehicles from a Central Depot to a Number of Delivery Points*. Operations research, 12(4):568-581

[25] **Croes, G.A.**: 1958, *A method for solving traveling salesman problems*. Operations Research, 6: 791–812.

[26] **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.

[27] **Bullnheimer, B.**, **Hartl, R.F.**, **Strauss, C.**: (1999); *An Improved Ant System Algorithm for the Vehicle Routing Problem*. Annals of Operations Research, 89, 319–28

[28] **Bell, J.E.**, **McMullen, P.R.**: (2004); *Ant Colony Optimization Techniques for the Vehicle Routing Problem*. Advanced Engineering Informatics 1;8, 41-48

[29] **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

[30] **Stützle, T.** and **Hoos, H.H.**: (2000); *MAX–MIN ant system*, Future Generation Computer Systems, 16(8): 889-914

[31] **Christofides, N.** and **Eilon, S.**: (1969); *An algorithm for the vehicle dispatching problem*. Journal of the Operational Research Society, 20: 309–318

[32] **Gillett, B.E.**, **Johnson, J.G.**: (1976); *Multi-terminal vehicle-dispatch algorithm*. Omega, 4(6):711-718

[33] **Chao, M.I.**, **Golden, B.L.** and **Wasil, E.A.**: (1993). *A new heuristic for the multi-depot vehicle routing problem that improves upon best known solutions*. American Journal of Mathematical and Management Sciences, 13(3-4):371-406

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