Journal of Computers Vol. 27, No. 4, 2016, pp. 41-54
doi:10.3966/199115592016122704004

Hybrid Ant Colony Optimization Algorithm for Solving the Open Vehicle Routing Problem

Bin Ge[1*], Yue Han[1], and Chen Bian[1]

School of Computer Science and Engineering, Anhui University of Science & Technology,
Huainan, Anhui 232001, China
bge@aust.edu.cn, 526774363@qq.com, yehuiliusu@yahoo.cn

Received 29 May 2015; Revised 11 August 2015; Accepted 28 August 2015

Abstract. Open vehicle routing problem (OVRP) is a hot research topic in modern operational research, compared with classical VRP problems, one of its marked characteristics is that the vehicle can choose the other distribution center as an end after the completion of the transportation service. The solving goal of OVRP is to build a Hamiltonian path to meet the needs of all customers. In order to solve the OVRP, a hybrid ant colony optimization (HACO) algorithm based on random distribution of loading and dynamically encoding was proposed. Firstly, the initial solutions were obtained through the method of random loading, and the colony optimization algorithm was adopted to get the optimal solution. Then the optimal solution was encoded as the zeroth particle of particle swarm algorithm. The initial fitness values were regarded as the historical optimal solution for individual. In order to get the best historical of individual and global optimal solution, the global optimal solution and the switching sequence of each particle was calculated and implemented, combining the hill climbing strategy for local search with side step. Computer simulations on the benchmark problems show that it can quickly and effectively get the known optimal solution or approximate solution.

Keywords: ant colony optimization algorithm, dynamic coding, open vehicle routing problem, particle exchange sequence, random stowage

1 Introduction

Open vehicle routing problem (OVRP) is expanded from the classical vehicle routing problem (VRP). The main difference between OVRP and VRP is: the vehicle route in the VRP is a Hamilton circuit (Hamiltonian tour). However, it is a Hamilton path in the OVRP (Hamiltonian path). It is mainly due to the following reasons: In the OVRP, it doesn't require the vehicles to return to the distribution center, or it's not necessary for the vehicles to return to distribution center after serving the last customers, even if they do, the vehicles must make the same trip in the reverse order. When the vehicle must return to the original starting point after the completion of the transportation task, this type of VRP is called a closed type; when the vehicle doesn't need to return to the original starting point after the completion of the task, this type of VRP is called an open type (see Fig. 1). The vehicle route of OVRP is open type rather than closed type. It is not necessarily to get the best Hamiltonian path by only removing one edge of the minimum Hamiltonian cycle. While solving the minimum Hamilton path is a NP (hard problem), the OVRP is also a NP. OVRP widely exists in transportation management, and it has many implications in real life, especially, it has great application value in the distribution service which has the characteristics of outsourcing business, such as campus shuttle bus issues, milk delivery issues, newspaper delivery issues, etc., in these kinds of problems, the distribution business would be outsourced to other vehicles or other teams because the enterprise does not have its own vehicles, and the enterprise does not require the vehicle to return to distribution center after serving the last customers.

* Corresponding Author

(a) Closed type of VRP (b) Open type of VRP

Fig. 1. Comparison between closed type of VRP and open type of VRP

In recent years, many international and domestic academics conduct the research for vehicle dispatch problem [1-2] and in particular for OVRP. Repoussis et al. [3] proposed a hybrid evolutionary algorithm for solving the OVRP, and simulate the known benchmark problem. The results show that this algorithm is superior to some known optimal solution in some instances. Emmanouil and Chris [4] proposed a heuristic algorithm based on new local search to solve the OVRP by searching for the neighborhood of a feasible solution, and recoded the local search operation into a static operation entity (SMD). Wah [5] established a mathematical model on the basis of asymmetric diagram, and design a neighbor particle swarm algorithm to solve the model, by using the model to solve the open vehicle routing problem with soft time windows, the simulation results show this method could obtain the feasible solution quickly. Fung et al. [6] put forward a kind of cultural genetic algorithm aimed at the open vehicle routing problem with limited arc. This algorithm could be used to find a set of open routes to tailor to mission requirements and minimize the cost. The experimental results show that the algorithm could obtain a better optimal solution than the traditional genetic algorithm. Lopez-Sanchez et al. [7] converted the distribution problem in real life into open vehicle routing problem by modeling. In order to minimize the time everyone spend on the vehicle, a kind of multiple starting point algorithm was proposed, the experimental results show that this algorithm could obtain high quality solutions within reasonable time. Marinakis and Marinaki [8] proposed an improved bee mating optimization algorithm using the process of local search to analog the motion of drone outside the honeycomb, the algorithm applied to two groups of numerical examples, the results show that this algorithm could obtain optimal solution in most cases.

The above algorithms explore the solution to OVRP in a certain extent, such as heuristic algorithm based on new local search, cultural genetic algorithm, multiple starting point algorithm, etc., however, these researches only pay attention to the improvement of the single algorithm, and it is difficult to avoid its inherent defects. For example, local search over-reliance on the search in local area and its adjacent domain, it is difficult to obtain the optimal solution in the global scope. Cultural genetic algorithm is a combination of population-based global search and local heuristic search based on individual, and this algorithm proposed a framework, under this framework, different search strategies could be used, therefore, the adopted strategy would affect the performance of the algorithm directly. This paper would synthetically consider the advantages and disadvantages of ant colony algorithm and particle swarm optimization algorithm and combine the advantage of ant colony algorithm such as initialization pheromone distribution, parallelism and positive feedback mechanism etc., so that the optimal solution could be obtained in the process of search, the global optimal path could be obtained further by take advantage of the global convergence of particle swarm algorithm and fast stochastic search. Compared with the heuristic algorithm introduced previous, the join of the improved particle swarm algorithm not only speeds up the convergence speed of the algorithm, but also enhances the evolution characteristic of the particle. Compared with the traditional pheromone update rule, this paper chooses three different best solutions to update pheromone, which makes the most of the optimal solution. The solve efficiency of the algorithm was improved dynamically by adjusting the number of times of local search. The optimal solution obtained by ant colony optimization algorithm would be encoded as the initial particle swarm of particle swarm optimization algorithm, and a kind of stochastic dynamic hybrid ant colony optimization (HACO) algorithm was proposed to solve the OVRP. The simulation results show that excellent results could be obtained.

The organization structure is as follows. The second part of this paper gives the mathematical model of OVRP; the third part proposes the hybrid ant colony algorithm in detail; the fourth part describes the process of the hybrid ant colony algorithm in detail; the fifth part compares the quality of the proposed algorithm with other algorithms in previous literature; the sixth part summarizes the full text and gives the direction of the future research.

2 Mathematical Model

2.1 Description of the problem

OVRP generally can be described as follows: there is a yard and a limited number of the vehicles with known capacity and transportation cost. At the same time, there is a set of customers with certain requirements, and the cost of transportation between each client, customers and yard transportation is known. The problem is to determine a vehicle route in order to minimize the following two objectives:

(1) The number of vehicles serviced for all customers.

(2) The total cost of the vehicle transportation.

It is generally accepted that the increased cost of the increased number of vehicles is always more than the cost saved by shortening the total distance, so the minimum number of vehicles and the minimum cost are generally considered as the first and the second optimization goals.

2.2 Mathematical model

OVRP is defined on a graph Z where L_{opt} is the vertex set and vertex, 0 represents the central depot and the remaining n-1 vertices represent the customer set. $L_{local} < L_{opt}$ is the arc set, $L_{local} = L_{opt}$ represent a customer, $L_{local} < L_{global}$ represent the set of delivery vehicles. $L_{global} = L_{local}$ represents the carrying load of the vehicle in the position $\tau_{\mathrm{ij}}(t+1) = G_{m\max}[(1-\rho)\cdot\tau_{\mathrm{ij}}(t) + \rho\cdot\Delta\tau(t)]$, and $N < N_{\max}$, $(X_1, X_2, ..., X_F)$ is associated with a cost which corresponds to the cost for transiting from $\mathrm{Min}\, z = \sum_{i,j} \mathrm{c}_{\mathrm{ij}} \sum_{k} x_{ijk} + R\cdot\sum_{k=1}^{m}\max\left(\left(\sum_{i=1}^{n} g_i y_{ik}\right) - q, 0\right)$ to P_{best}, P_g represents the Distribution cost matrix between node-customer set, v represent the distance from customer P_{best} to customer P_g. If the vehicle $T < T_{\max}$ transferred from customer i to customer j, $x_{ijk} = 1$, or else, $x_{ijk} = 0$; $i, j = 1, 2, ..., n$, v.

The optimization model of open vehicle routing problem can be given as follows:

$$\mathrm{MinM}. \tag{1}$$

$$\mathrm{Min} \sum_{i,j} \mathrm{c}_{\mathrm{ij}} \sum_{k} x_{ijk} \tag{2}$$

Constraint condition:

$$\sum_{k=1}^{m} y_{jk} = \begin{cases} 1 & , i = 1, 2, ..., n \\ M & , i = 0 \end{cases} \tag{3}$$

$$\sum_{i=1}^{n} q_i y_{ik} \le Q, k = 1, 2, ..., M \tag{4}$$

$$\sum_{j=1}^{n} x_{ijk} = \sum_{j=1}^{n} x_{jik} = y_{ik}, i = 1, 2, ... n; k = 1, 2, ..., M \tag{5}$$

$$y_{ik} \in \{0, 1\} \tag{6}$$

$$y_{ik} \in \{0, 1\} \tag{7}$$

Count of the minimize vehicle

$$m_{\min} = \left\lceil (\sum_{i=1}^{n} q_i) / Q \right\rceil \quad \textbf{(8)}$$

Thereinto, the objective function (1) is first target and minimize total vehicle travel numbers, the objective function (2) is second target; constraints (3) ensure that each customer is visited by exactly one vehicle; Constraints (4) are the capacity constraints for the vehicles. Constraints (5) ensure that each vehicle leave immediately after visiting a customer point Constraints (6) and (7) represent the ranges of parameters.

3 HACO

Ant colony algorithm and particle swarm algorithm belong to the typical swarm intelligence algorithm, each algorithm has its advantages and disadvantages, basic ant colony algorithm has slow convergence speed, and it is easy to fall into local optimum, the initial pheromones are scarce in ant colony algorithm. The particle swarm is easy to produce premature convergence, and its local search ability is poor. The study of swarm intelligence algorithm work could mainly divided into two categories: the first is to combine different swarm intelligent algorithm in order to improve the performance of a single group of intelligent algorithm, it is the main tendency of the research, the second is to combined local optimization algorithm with global optimization algorithm. Therefore, it could improve the quality and the operational efficiency of the solution to the ant colony algorithm significantly while two algorithms merge together and become a HACO algorithm.

The HACO algorithm proposed in this paper, first of all, using a random loading method to generate the initial solution set, while all the ants traversed and build the solution, HACO use three different optimal solutions to update the pheromone matrix, it could prevent the optimal ant colony algorithm from converges to global minimal. And then it would encode the optimal solution obtained by ant colony optimization algorithm as the zeroth particles of particle swarm optimization algorithm, and the initial value would be an individual historical optimal solution, the global optimal solution and the exchange sequence v of each particle would be calculated. Secondly, aiming at the advantages of particle swarm optimization algorithm while solving problems, the algorithm swapping the position of each particle sequence, in order to get a new location, the improved particle swarm algorithm not only speeds up the convergence speed of the algorithm but also enhance the evolution characteristic of the particle, at last it use side step climbing strategy for local search, individual historical optimal solution and the global optimal solution would update if there are improvements, through this search mode, searching performance of particle swarm optimization algorithm and the global convergence will improve. Combined ant colony algorithm with particle swarm optimization algorithm, on the one hand, the searching efficiency of particle swarm optimization would improve; on the other hand it makes the ants obtain the characteristics of the "particles."

3.1 Selection of the path

This algorithm selects the deterministic search and exploratory search as the path selection rules, the state transition rules of ant [9] as shown in the following type:

This algorithm uses the pseudo-random proportion rule to choose the next path to visit clients:

$$j = \begin{cases} \arg\max_{j \in N_i^k} \{\tau_{ij}\}, & q \le q_0 \\ J, & q > q_0 \end{cases} \quad \textbf{(9)}$$

$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}(t)}{\sum_{j \in N_i^k} \tau_{ij}(t)}, & j \in N_i^k \\ 0, & \text{else} \end{cases}$ represents the customers selected by using the method of roulette wheel:

$$p_{ij}^{k}\left(t\right)=\begin{cases}\dfrac{\tau_{ij}(t)}{\sum\limits_{j\in N_i^k}\tau_{ij}(t)}, j\in N_i^k\\ 0, \qquad \text{else}\end{cases} \quad \textbf{(10)}$$

Thereinto: s_b^r represents the size of the pheromone on the edge of s_b^s at time t. $\Delta\tau_{\mathrm{ij}}=\gamma_1\delta(s_b^i,(i,j))$ $+\gamma_2\delta(s_b^r,(i,j))+\gamma_3\delta(s_b^s,(i,j))$ represents the probability of the transfer of the ants γ_1 moving from node γ_2 to node γ_3 at time t. $\gamma_i\geq 0,(i=1,2,3)$ represents the set of customer the ant γ_1 at node $\gamma_1+\gamma_2+\gamma_3=1$ was allowed to select at the next time. $\delta(x,(i,j))=\begin{cases}1, \text{solution of } x \text{ includes } (i,j)\\ 0, \text{otherwise}\end{cases}$ represents the random variable of uniform distribution in the range. $G_{m\max}=\begin{cases}\tau_{\min}, x<\tau_{\min}\\ x, \quad \tau_{\min}\leq x\leq\tau_{\max}\\ \tau_{\max}, x>\tau_{\max}\end{cases}$ represents the choice probability value changed dynamically in the volution, control algorithm keep balance between centralized search and diversified search, the initial value is 0.5. When $\rho(t+1)=\begin{cases}\max[\lambda\cdot\rho(t),\rho_{\min}], n_c=n_{\max}\\ \rho(t), \text{ otherwise}\end{cases}$, the selective probability in the formula (10) decides the next customers to visit, and diversify search would be operated, else the centralized search would be operated.

This method can decrease the influence caused by the positive feedback of ant colony algorithm, and making the selection of the customers directional, the search could avoid algorithm from trapping in local optimal solution because of its diversity.

3.2 Pheromone updating

When an iteration is end, the pheromone matrix would update after all the ants construct solution, in order to enhance the pheromone concentration on the edge of the optimal solution, it makes the ants selected by the ants with larger probability in later iterations, we use three different best solutions to update the pheromone:

(1) s_b^i : When all the ants find the best solution in the current iteration;

(2) s_b^r : The best solution the ants found since the resumption;

(3) s_b^s : The global optimal solution found since its launch.

The pheromone in the edge of released could be calculated according to following formula [9]:

$$\Delta\tau_{\mathrm{ij}}=\gamma_1\delta(s_b^i,(i,j))+\gamma_2\delta(s_b^r,(i,j))+\gamma_3\delta(s_b^s,(i,j))\,. \quad \textbf{(11)}$$

Among γ_1, γ_2, and γ_3, respectively represent the weights of three different kinds of best solution in update of the pheromone, three weighting parameters meet the following conditions: $\gamma_i\geq 0$, $(i=1,2,3)$ and $\gamma_1+\gamma_2+\gamma_3=1$.

The definition of $\delta(x,(i,j))$ as follows:

$$\delta(x,(i,j))=\begin{cases}1, \textit{The solution x containsedge } (i,j)\\ 0, \textit{else}\end{cases} \quad \textbf{(12)}$$

The update rule of the pheromones as follows:

$$\tau_{\mathrm{ij}}(t+1)=G_{m\max}[(1-\rho)\cdot\tau_{\mathrm{ij}}(t)+\rho\cdot\Delta\tau(t)]\,. \quad \textbf{(13)}$$

$$G_{m\max} = \begin{cases} \tau_{\min}, x < \tau_{\min} \\ x, \quad \tau_{\min} \le x \le \tau_{\max} \\ \tau_{\max}, x > \tau_{\max} \end{cases} \quad \textbf{(14)}$$

$$\rho(t+1) = \begin{cases} \max[\lambda \cdot \rho(t), \rho_{\min}], n_c = n_{\max} \\ \rho(t), \ else \end{cases} \quad \textbf{(15)}$$

Among them, ρ is the residual factor to pheromones, $\Delta\tau_{ij}$ represents the increment of the pheromones the ants created on the edge of (i, j) in $[t, t+1]$, n_c represents continues no-evolutionary cycle times, $n_{\max}$ is a constant, $\lambda \in (0,1)$ is a constant and control the attenuation speed ρ, $\rho_{\min}$ is the minimum value ρ, prevent ρ from being too small so the convergence speed would be affect. When n_c reach a preset value $n_{\max}$, ρ would decrease, and n_c recount, repeatedly, until ρ reaches the minimum value $\rho_{\min}$.

3.3 Construction of initial solution set

Constructing initial solution is the first step in the algorithm; a good initial solution helps the algorithms converge to the optimal solution quickly. The initial solution set is generated by a method of random loading vehicle [10], specific steps are as follows:

Step 1: First of all, the customer sequence $C = \{c_1, c_2, ..., c_n\}$ could be obtained according to the length of the distance between each customer and distribution center and sorted from small to large, then calculated $\overline{F}_1$ by the formula (16), and initialize the code of the path, $CarNum = 0$.

Step 2: determining whether the customer sequence is empty, if not, then $p = 1$, or else end it.

Step 3: put the customer c_i whose number is p into the path $CarNum$, and judging whether it satisfied the constraint condition, if do, c_i would be added into the path $CarNum$, and delete c_i from C, $n = n-1$, $p = p + rand(0, 2\times\overline{F}_1 - 2\times CarNum)$; or else $p = p+1$.

Step 4: determining whether p is greater than n, if p is greater than n, then the path $carInit$ would be added into the initial solution concentration, $CarNum = CarNum + 1$ skip to Step 2, otherwise, skip to Step 3.

$Rand(0, 2\times\overline{f}_1 - 2\times CarNum)$ represents an integer randomly generated distributed between 0~$rand(0, 2\times\overline{f}_1 - 2\times CarNum)$ uniformly, $\overline{F}_1$ is the estimated number of vehicles obtained by the following formula:

$$\overline{F}_1 = \left\lceil \frac{\sum_{i\in V} d_i}{\alpha Q} \right\rceil \quad \textbf{(16)}$$

Thereinto, α represents the load coefficient, which is used to estimate the load level of the vehicle teams, usually, more constraint conditions mean that vehicle can load less goods, the corresponding α is lower.

In the process of construct the initial solution set, α is a random value. Using the random loading method to generate the initial solution set of the vehicle, not only making each solution exists in the feasible region, but also ensure the initial solution could be uniform distribution by random function.

3.4 Local search

A hill climbing strategy is adopted to improve the local search in this paper [11]; the strategy is based on the Pareto dominance relationship. The decline directions or non- dominant direction is constructed according to the distance of the solution system. The local search schematic diagram is shown in Fig. 2.

In Fig. 2, $\{-,-\}$ represents the falling cone of particle x_0, any x in this area satisfies $x > x_0$. $\{+,+\}$

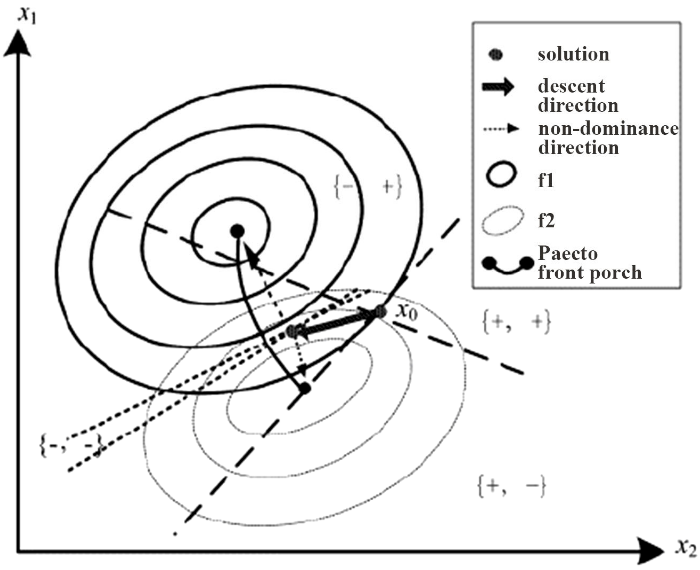

Fig. 2. The hill climbing strategy

represents the rising cone of particle x_0. Points in this area were dominated by x_0. When x is located in multiple cone $\{-,+\}$, $f_1(x) < f_1(x_0)$ and $f_2(x) > f_2(x_0)$. When the dimension is greater than or equal to three-dimensional, the rest may be deduced by analogy. When using the local search strategies of particle:

(1) Early particle swarm is generally far away from the front. Falling cone and rising cone each accounted for half of the space, due to the optimal dominance solutions is easy to find in the cone, so using falling direction to guide the particle swarm to the fast convergence of the frontier. Falling cone uses a ladder method in order to solving more common practical problems. $d = x - x_0$ is the descent direction of x_0 in $\{-,-\}$, $d = x_0 - x$ in area $\{+,+\}$.

When the dominance solution obtained by the limited local search are uniform within M_{ls} times, it shows that the particle swarm is very close to the front, at this moment, the multiple cone is far greater than falling cones and rising cones, the search direction should be changed, the guided particle swarm diffused with the goal of diversification in the front, the dominant direction as shown in Fig. 2, expressions for calculation is:

$$\overline{d}_s = \sum_{i=1}^{M_{ls}} ((x_i - x_0) / \|(x_i - x_0)\|) \tag{17}$$

where $x_i(1,...,M_{ls})$.

(2) According to the relationship between the number of the searched non-dominance solutions and N_{ls}, the distance between the particles and frontier could be judged and the search direction could be choose adaptively. When the number of the non-dominance solutions is greater than or equal to N_{ls}, in accordance with the second case to deal with, otherwise in accordance with the first one to deal with.

The local search algorithm based on the Pareto not only calculate fast, but also can find a group of great local optimal solution, yet the results largely depend on the selection of initial point, therefore, some improvements were made in this paper, the original single neighborhood search scope change into multiple neighborhood. Due to the expansion of the search scope, the algorithm finally got a huge set of local optimal solution, and reduced the dependence of the solution set on initial solution.

The basic steps of the improved algorithm are as follows:

Assuming set of initial solution were S_1, the searched best solution set is S_B, assuming $N(s)$ is the neighborhood of feasible solution set S, P is the empty set.

Step 1: Obtain any one from S_1.

Step 2: Calculate $N(s)$ and reserve the weaker solutions;

Step 3: Compared the solution $S_B \cup S_I$ obtained in with the solution obtained in Step 2, and preserved the weak solutions in P;

Step 4: Compare P with each solution in S_I, and retain the branch solution in S_I;

Step 5: $(S_I \setminus \{s\}) \bigcup P \longrightarrow S_I$.

Step 6: Compare P with each solution in S_B, and retain the branches solution in S_B;

Step 7: Determine whether S_I is the empty set, repeat Step 1 to Step 6 if it is empty, otherwise S_B is saved as a prayer.

3.5 Particle swarm optimization algorithm

Encode particle. In order to apply particle swarm algorithm [12] into OVRP, particles should be encoded. *i* represents the customer, 0 represents distribution center, insert m-1 zero into the route contained *n* clients, *m* is the vehicle number, route sequence is divided into *m* segments, each particle is a $n+m$-1 dimensional vector.

Assuming in an OVRP, there are 3 car and 8 customers in distribution center, one of the particles is: 2 1 0, 3 0 6 8 5, the access sequence of three cars as shown below:

Vehicle 1: distribution center - 2-1.

Vehicle 2: distribution center - 4-3.

Vehicle 3: distribution center - 7-6-8-5.

The relationship between the particles of Vehicle number m and customer number n and the position X of dimensions vector D is $D = m + n - 1$, the position vector X of the particles and the velocity vector V share the same dimension.

Because the particles arrive to a new place, there may be two adjacent 0, and the beginning of the particles and the end of it are 0, in order to improve the efficiency, it is necessary to estimate before calculating the fitness value, if the two cases happened, the fitness value would be set as infinity.

Evolution equation of particle swarm. Particle swarm optimization (PSO) algorithm is based on groups; the individual of the groups move to the better area according to the fitness of the environment. By the iteration t, particle i in the D space search dimensions is expressed as $X_i^t = (X_{i1}^t, X_{i2}^t, ..., X_{iD}^t)$, the best location it experienced (with the best fitness) is $P_i^t = (P_{i1}^t, P_{i2}^t, ..., P_{iD}^t)$, also known as P_{best}; the best location of all particles with the index number in group experienced is represented by symbolic g, is P_g, also known as g_{best}. The speed of the particles is represented by $V_i^t = (V_{i1}^t, V_{i2}^t, ..., V_{iD}^t)$ [13-14]. Basic PSO evolution equation is shown below:

$$v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_1 (p_{ij}^t - x_{ij}^t) + c_2 r_2 (p_{gj}^t - x_{ij}^t) \quad \textbf{(18)}$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad \textbf{(19)}$$

Among them *j* represents the j dimension of the particle, *i* represents particle i, represent the generation, and *t* represent the iteration t, the values are always between the [0, 2], r_1 and r_2 represent a random number between [0, 1], the two number are uncorrelated. ω is inertia weight, represent the influence the previous variable quantity of the speed made on the current level, maximizing the ω could enhance the global search ability [15], while the smaller value is beneficial to improve the local search ability, speed up the convergence, so the algorithm use large inertia weight value in early and use smaller inertia weight value in late, it could foster strengths and circumvent weaknesses and not only improve the global search ability, but also guarantee the convergence efficiency. The computation formula is as follows:

$$\omega = \omega_{\max} - StepNow \times \frac{\omega_{\max} - \omega_{\min}}{Step} \quad \textbf{(20)}$$

In the last-written denominator $\omega_{\max}$ represents the maximum inertia weight, $\omega_{\min}$ represents the minimum inertia weight, Step represent the total number of iterations, StepNow represent the current number of iterations.

In order to apply the PSO algorithm to the vehicle routing problem, according to the method of transformation in literature [16], introduce the concept of abelian groups and abelian subgroups, after modified, the formula (18) and formula (19) as shown below:

$$v_{ij}^{t+1} = \omega v_{ij}^{t} \oplus rand_1\varphi_1 \cdot (p_{ij}^{t} - x_{ij}^{t}) \oplus rand_2\varphi_2 \cdot (p_{gj}^{t} - x_{ij}^{t}) \quad \textbf{(21)}$$

$$x_{ij}^{t+1} = x_{ij}^{t} \oplus v_{ij}^{t+1} \quad \textbf{(22)}$$

4 HACO Algorithm Flow

Step 1: set the number of ants m, set the parameters ω, c_1, and c_2 of particle swarm, and initializes the speed and position of each particle.

Step 2: initialize the pheromone of each side, put m ants in n cities randomly.

Step 3: use random loading vehicle method to generate the initial solution set.

Step 4: calculate the length L_{local} of the path every ant get, and the local optimal solution is $L_{local} = \sum_{k=1}^{m} L_k$, and record the path correspond to local optimal solution.

Step 5: apply the variable neighborhood descent

procedure to local optimal solution L_k [17], 2 - opt, the times of the local search is, the initial value could be bigger so that the solution space could be searched sufficiently, the value decreases with the process of the evolution. If the new path is optimal and satisfy the vehicle load constraint, the optimization solution L_{opt} and the optimal path would be saved. When $L_{local} < L_{opt}$ updating the optimal path, make $L_{local} = L_{opt}$.

Step 6: compare L_{local} with L_{global}, if $L_{local} < L_{global}$, then make $L_{global} = L_{local}$, and updating the optimal path at the same time.

Step 7: according to formula (13), update the global pheromone.

Step 8: if do not meet the termination conditions, and then back to Step 2.

Step 9: the optimal solution of ACO algorithm would be serve as the zeroth particle, and as a template, the remaining components $(X_1, X_2, ..., X_F)$ would be generated randomly, the fitness of each particle value calculated according to the formula:

$$\text{Min } z = \sum_{i,j} \mathrm{c}_{\mathrm{ij}} \sum_{k} x_{ijk} + R \cdot \sum_{k=1}^{m} \max\left(\left(\sum_{i=1}^{n} g_i y_{ik}\right) - q, 0\right)$$

Step 10: adapt the initial fitness value P_{best} as an individual historical optimal solution, and calculate the global optimal solution P_g.

Step 11: calculate abelian ordered group v of each particle.

Step 12: switching sequence of each particle according to formula (19), and obtain the new location.

Step 13: calculate each particle's fitness, combining the hill climbing strategy for local search with side step, compared P_{best} with P_g the existing and, if there are improvements, then update the fitness.

Step 14 judge whether the algorithm meets the termination conditions, if meet then output the optimal solution, otherwise, jump to Step10.

The pseudo codes are as follows as follows:

```
Start
Initial parameters c1=1.7, c2=1.7, ωmax=0.9, ωmin=0.4, ω=0.6, and m=60;
Do N=0, N=N+1;
use random loading vehicle method to generate the initial solution set Llocal,
calculate the length Llocal of the path every ant get, and the local optimal so-
```

```
lution is $L_{local}=\sum_{k=1}^{m}L_k$ , and record the path correspond to local optimal solution;
  apply the variable neighborhood descent procedure to local optimal solution $L_k$ until search time more than $Z$;
  If the new path is optimal and satisfy the vehicle load constraint, the optimization solution $L_{opt}$ and the optimal path would be saved;
  If $L_{local}<L_{opt}$, updating the optimal path, set $L_{local}=L_{opt}$;
  If $L_{local}<L_{global}$, $L_{global}=L_{local}$;
  Utilizing $\tau_{ij}(t+1)=G_{m\max}[(1-\rho)\cdot\tau_{ij}(t)+\rho\cdot\Delta\tau(t)]$, update the global pheromone. While $N<N_{\max}$;
  the optimal solution of ACO algorithm would be serve as the zeroth particle, and as a template, the remaining components $(X_1,X_2,...,X_F)$ would be generated randomly;
  Do T=0, T=T+1;
  the fitness of each particle value calculated according to the formula:
```

$$\text{Min } z=\sum_{i,j}\mathrm{c}_{ij}\sum_{k}x_{ijk}+R\cdot\sum_{k=1}^{m}\max\left(\left(\sum_{i=1}^{n}g_i y_{ik}\right)-q,0\right)$$

```
  adapt the initial fitness value $P_{best}$ as an individual historical optimal solution, and calculate the global optimal solution $P_g$;
  calculate abelian ordered group $v$ of each particle;
  compared $P_{best}$ with $P_g$ the existing and, if there are improvements, then update the fitness;
  While $T<T_{\max}$;
  Output optimal solution.
  End
```

5 Experimental Results and Analysis

In this paper, the simulation experiment platform environment is 2.10 GHz PC processor Intel Celeron (R) (R) G540T CPU and 4.0 GB memory. Matlab7.0 is used to programming. The initializations of HACO simulation parameters are as follows:

Table 1. The initializations of HACO simulation parameters

Particle swarm algorithm	Ant colony algorithm	
$\omega_{\max}=0.9$	$\gamma_1=0.2$	$q_0=0.5$
$\omega_{\min}=0.4$	$\gamma_2=0.4$	$n_{\max}=100$
$\omega=0.6$	$\gamma_3=0.4$	$m=0.6*n$
$c_1=1.7$	$\tau=10^{-6}$	$Q=100$
$c_2=1.7$	$\lambda=0.25$	$\rho=0.25$

In order to evaluate the proposed HACO algorithm, 14 benchmark problem proposed by Christofides [18] and eight large open vehicle routing problem proposed by Li et al. [19] were simulated. In the benchmark problems proposed by Christofides, the total number of all the nodes includes distribution nodes are between 51 and 200. The position of the node is in Cartesian rectangular coordinate system; assume the transportation cost from the node to as their Euclidean distance. Every problem has load constraint; problem6-10, 13, and 14 also has the maximum path length constraint. Top 10 benchmark problem distributed within a square unit randomly, the remaining problems scattered distribute with no rules, and regional distribution center is not in the central region. In the benchmark problems proposed by Li, the total number of nodes including distribution centers are between 200 and 480, besides, there are load constraints.

Table 2 and Table 3 show the simulation results aim at the solution to the benchmark problem proposed by Christofides, each sub-problem was tested 10 times, including the optimal value, average value and standard error. BKS1is the first optimal solution, and it minimize the use number of vehicles at first, then minimize the total distance traveled, BKS2 is the second optimal solution, and it only minimize the total distance. Generally thinking, the cost of a car added every time, total more than total distribution cost savings, so regard minimizing the vehicle number as the first goal.

Table 2. The comparison between HACD and BSK1

benchmark problem	BKS1	NV	HACO	Error of HACO	Average result of HACO
C1	416.06	5	**416.06**	0	416.28
C2	567.14	10	**567.14**	0	567.31
C3	639.74	8	**639.74**	0	640.21
C4	733.13	12	734.36	0.17	735.60
C5	893.39	16	894.24	0.01	895.61
C6	412.96	6	**412.96**	0	413.19
C7	583.19	10	**583.19**	0	583.50
C8	644.63	9	**644.63**	0	644.93
C9	757.84	13	759.47	0.22	759.61
C10	875.67	17	877.28	0.18	879.59
C11	682.12	7	**682.12**	0	682.38
C12	534.24	10	535.28	0.19	535.49
C13	904.04	11	**904.04**	0	904.36
C14	591.87	11	592.16	0.05	592.44

Table 3. The comparison between HACD and BSK2

benchmark problem	BKS2	NV	HACO	Error of HACO	Average result of HACO
C1	412.96	6	412.96	0	413.16
C2	564.06	11	564.06	0	564.26
C3	639.26	9	639.26	0	639.90
C5	869.00	17	870.47	0.17	872.78
C7	568.49	11	568.95	0.01	569.24
C9	756.38	14	757.24	0	757.56
C11	678.54	10	678.54	0	678.79
C13	896.50	12	897.10	0	897.41

By the standard error of Table 2 and Table 3, it is clear that the proposed algorithm HACO share the much same optimal solution with it in other documents. The whole population of particles iterates 500 times and operates 50 times randomly; its standard error will be obtained. If the using number of the vehicles were minimized at first, then minimize the total distance traveled, the optimal solution of the standard error is between 0 and 0.22; when only to minimize the total distance traveled, the standard error of the optimal solution is between 0 and 0.17; Fig. 3 is an algorithm simulation diagram of benchmark problem C3 HACO. In Fig. 3, each dot represents a customer, the initialization of the number of ants and the pheromone of each side is according to HACO, the initial solution set iterates 500 times, the optimal solution of each generation will be reserved.

Table 4 represent the simulated experimental results of the benchmark problem proposed by Christofides and Li and Table 5 represent the results of other algorithms. By comparing the optimal solution, it is clear that 10 known optimal solutions obtained by HACO algorithm proposed in this paper are better than the optimal solutions obtained by the other two algorithms, on the premise of the same minimized number of vehicles used, the vehicles travel shorter distance. However, in some individual benchmark problems such as C10, the performance of HACO algorithm lagged behind the other two algorithms.

In view of the open vehicle routing problem, the advantages and disadvantages of the ACO and PSO algorithm are combined, a hybrid ant colony optimization algorithm (hybrid ant colony optimization, HACO) is proposed.

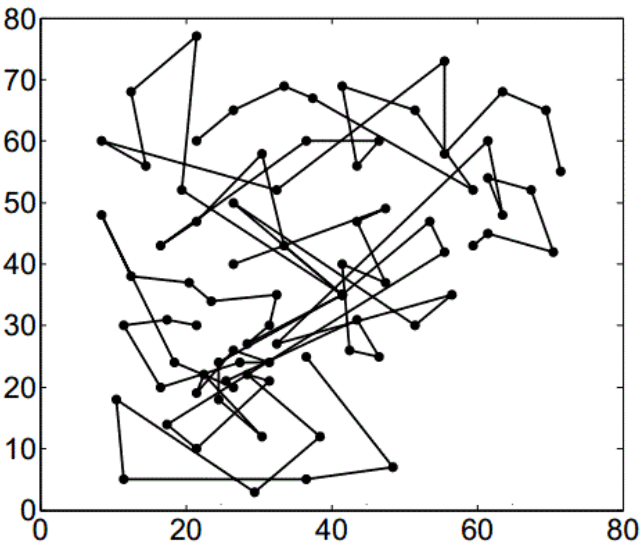

Fig. 3. HACO algorithm simulation diagram of benchmark problem C3

Table 4. The comparison between HACD and other algorithm

benchmark problem	HybPSO [20]		HES [21]		HACO	
	TD	NV	TD	NV	TD	NV
C1	**416.06**	5	**416.06**	5	**416.06**	5
C2	**567.14**	10	**567.14**	10	**567.14**	10
C3	**639.74**	8	**639.74**	8	**639.74**	8
C4	735.29	12	**733.13**	12	735.18	12
C5	895.79	16	894.11	16	**893.39**	16
C6	**412.96**	6	**412.96**	6	**412.96**	6
C7	**583.19**	10	584.15	10	**584.15**	10
C8	644.79	9	**644.63**	9	**644.63**	9
C9	759.81	13	764.56	13	759.61	13
C10	878.49	17	888.46	17	879.59	17
C11	**682.12**	7	**682.12**	7	**682.12**	7
C12	535.49	10	**534.24**	10	534.24	10
C13	**904.04**	11	910.26	11	**904.04**	11
C14	592.58	11	**591.87**	11	592.44	11

Note. Table based on benchmark problem proposed by Christofides.

Table 5. The comparison between HACD and other algorithm

benchmark problem	HybPSO [20]		HES [21]		HACO	
	TD	NV	TD	NV	TD	NV
O1	6023.25	5	**6018.52**	5	6021.11	5
O2	4557.89	9	4583.7	9	**4557.38**	9
O3	7734.28	7	7733.77	7	7735.14	7
O4	7268.23	10	7271.24	10	7267.18	10
O5	9201.28	8	9254.15	8	7267.18	8
O6	9797.28	9	9821.09	9	9798.19	9
O7	10352.29	10	10363.4	10	10351.18	10
O8	12419.25	10	12428.2	10	12418.57	10

Note. Table based on benchmark problem proposed by Li.

6 Conclusion

In this paper, a random loading method is used to get the initial solution set, then encoding the optimal solution of ant colony optimization algorithm as the initial solution of PSO algorithm. According to the simulation of benchmark problem, the results show that this algorithm can solve OVRP effectively compared with other heuristic algorithms. In the future, the uncertain factors of open vehicle routing problem and the aspects of multi-depot will be investigated.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (61070220, 61170060), Anhui Provincial Natural Science Foundation of China (1408085ME110), Anhui Provincial Major Project of Colleges and universities Natural Science (KJ2013ZD09) and Anhui Provincial Key Project of Colleagues and universities Natural Science (KJ2012A096).

Reference

[1] Y.C. Shi, L. Zhang, P. Liu, Survival analysis of urban traffic incident duration: a case study at Shanghai expressways, Journal of Computers (Taiwan) 26(1)(2015) 29-39.

[2] H.W. Wang, H.W. Chen, The impact analysis of patent litigation for mobile service interface technology via supply Chains in Taiwan, Journal of Computers (Taiwan) 23(4)(2012) 43-46.

[3] P. P. Repoussis, C.D. Tarantilis, O. Braysy, G. Ioannou, A hybrid evolution strategy for the open vehicle routing problem, Computers and Operations Research 43(3)(2010) 443-455.

[4] E.Z. Emmanouil, T.K. Chris, An open vehicle routing problem metaheuristic for examining wide solution neighborhoods, Computers and Operations Research 37(2010) 712-723.

[5] G.H. Sun, Open with soft time windows vehicle routing problem, Computer Engineering and Applications 4(17)(2011) 13-17.

[6] Y.K.R. Fung, R. Liu, Z.B. Jiang, A memetic algorithm for the open capacitated arc routing problem, Transportation Research 50(2013) 53-67.

[7] A.D. Lopez-Sanchez, A.G. Hernandez-Diaz, D. Vigo, R. Caballero, J. Molina, A multi-start algorithm for a balanced real-world open vehicle routing problem, European Journal of Operational Research 238(3)(2014) 104-113.

[8] Y. Marinakis, M. Marinaki, A bumble bees mating optimization algorithm for the open vehicle routing problem, Swarm and Evolutionary Computation 15(2014) 80-94.

[9] X.Y. Li, P. Tian, The open vehicle routing problem with ant colony optimization algorithm, Systems Engineering Theory & Practice 6(2008) 81-93.

[10] J. Wang, B. Li, A multi-objective tabu vehicle routing problem with fuzzy due time search algorithm, Computer Integrated Manufacturing Systems 17(4)(2011) 858-866.

[11] L.P. Wang, Q.H. Wu, F.Y. Qiu, Y.C. Wu, Local search and improved MOPSO hybrid optimization, Algorithm and Its Application in Computer Science 39(11)(2012) 174-200.

[12] Z. Huang, Hybrid quantum particle swarm optimization algorithm for vehicle routing problem, Computer Engineering and Applications 49(24)(2013) 219-223.

[13] T.S. Pan, T.K. Dao, T.T. Nguyen, S.C. Chu, Optimal base station locations in heterpgeneous wireless sensor network based on hybrid particle swarm optimization with bat algorithm, Journal of Computers (Taiwan) 25(4)(2014) 14-25.

[14] M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Information Sciences 294(2015) 182-202.

[15] D.F. Li, H.X. Guo, L.H. Liu, W.W. Li, Improved particle swarm optimization algorithm for vehicle route optimization, Computer Engineering and Applications 48(20)(2012) 216-223.

[16] W.F. Jiang, J.M. He, R. Wu, The vehicle routing problem of particle swarm optimization with reverse logistics management, Systems Engineering-Theory Methodology Application 17(4)(2008) 475-479.

[17] P. Chen, H.K. Huang, X.Y. Dong, A variable neighborhood solving vehicle routing problem search algorithm, Journal of System Simulation 23(9)(2011) 1945-1950.

[18] N. Christofides, A. Mingozzi. P. Toth, The vehicle routing problem, in: N. Christofides, A. Mingozzi, P. Toth, C. Sandi (Eds.), Combinatorial Optimization, Wiley, Chichester, 1979, pp. 315-338.

[19] F. Li, B. Golden, E. Wasil, The open vehicle routing problem: algorithms, largescale test problems, and computational results, Computers & Operations Research 34(2007) 2918-2930.

[20] Y. Marinakis, M. Marinaki, A hybrid particle swarm optimization algorithm for the open vehicle routing problem, in: M. Dorigo et al. (Ed.), ANTS 2012, Lecture Notes in Computer Science, Vol. 7461, Springer-Verlag, Berlin, 2012, pp. 180-187.

[21] P.P. Repoussis, C.D. Tarantilis, O. Braysy, G. Ioannou, A hybrid evolution strategy for the open vehicle routing problem, Computers & Operations Research 37(2010) 443-455.