

Improved Bat Algorithm for Vehicle Routing Problem

Yu Li^{a,b}, Qian Guo^b, and Jingsen Liu^{c,*}

^a*Institute of Management Science and Engineering, Henan University, Kaifeng, 475004, China*

^b*Business School, Henan University, Kaifeng, 475004, China*

^c*Institute of Intelligent Network system, Henan University, Kaifeng, 475004, China*

Abstract

Vehicle routing problem (VRP) is the key issue of logistics system optimization. As a classical combinatorial optimization problem, it belonged to the typical NP-hard problem and remained unsolved. In this paper, the novel bat algorithm is proposed to solve VRP. The improvement is based on the combination of dynamic inertia weight and time factor. It can take full advantages of dynamic search by the random velocity and random step-size. Furthermore, with the real-number encoding approach, the discrete VRP can be converted into a quasi-continuous one. The procedure of the optimal searching in multidimensional continuous space can be implemented directly. Experimental results indicate that improved bat algorithm performs well for vehicle routing problem.

Keywords: vehicle routing problem; bat algorithm; inertia weight; time factor; real-number encoding

(Submitted on October 23, 2018; Revised on November 24, 2018; Accepted on December 27, 2018)

© 2019 Totem Publisher, Inc. All rights reserved.

1. Introduction

Vehicle routing problem was first proposed by G. Dantzig and J. Ramser in 1959 [1]. Once proposed, it attracted experts and scholars from mathematics, operations research, computer and other fields to research. At present, VRP problem is mostly solved by algorithms, which mainly includes exact algorithms [2] (such as: Branch and Bound Approach, Dynamic Planes Approach, Cutting Planes Approach, etc.) and heuristic algorithms. The heuristic algorithms also include traditional heuristic algorithms like Nearest-Neighbor Method, Clarke-Wright Saving Algorithm, etc. and meta-heuristic algorithms. The meta-heuristic algorithms are mainly referred to intelligence optimization algorithms like Tabu Search [3], Genetic Algorithm [4], etc. When the problems' scale is constantly expanding, it is very difficult to find the optimal solution with the exact approaches. At present, the research on VRP by experts and scholars focuses on heuristic algorithms. The literature has proved that the standard vehicle routing problem is NP-hard combinatorial optimization problem, so the variant problems of different types of standard vehicle routing problems is also NP-hard combinatorial optimization problem. It is also challenging to construct more efficient algorithm for solving VRP.

In general, vehicle routing problems are described as follows: For a series of delivery points, a reasonable driving route is formed so that the vehicles pass through these points in an orderly manner. When meeting certain constraints (such as vehicle capacity, delivery or receiving time, distance limit, etc.), the driving route can reach some specific goals (such as the shortest driving route, the shortest time, the lowest cost, the highest profit, etc.). In practice, there are many different types of vehicle routing problems. Through the research and analysis of the literature in recent years, the vehicle routing problems can be divided into different types according to the different factors and constraints. According to whether there is a time window constraint, VRP can be divided into two types: the VRP with a time window (VRPTW) [5] and the VRP without a time window. Depending on the vehicle's affiliation to the car park, it can be divided into two types: Open VRP (OVRP) [6] and closed vehicle routing problems. According to the type of vehicle, the problems can be divided into homogeneous vehicle routing problems and heterogeneous vehicle routing problems [7]. According to the number of car parks can be divided into single depot vehicle routing problems and multiple depot vehicle routing problems [8]. With the in-depth study

* Corresponding author.

E-mail address: ljs@henu.edu.cn

of vehicle routing problem, many practical problems cannot be simply classified into a certain type, but more combinatorial problems that combine multiple constraints.

Bat Algorithm (BA) is a novel bionic algorithm proposed by Yang in 2010 [9]. This novel meta-heuristic optimization algorithm is inspired by the echolocation system of micro-bats. Bat algorithm has been widely concerned, because of its many advantages. For example, its algorithm model is simple, its parameter is less than the other algorithm, and it is easy to achieve. In recent years, many researchers have applied the bat algorithm to optimization problems and have made good progress. Bat algorithm shows better performance in many areas as follows: function optimization [10], image compression [11], combinatorial optimization [12], production scheduling [13], path planning [14] and so on. For all intelligence optimization algorithms, how to balance exploration and exploitation is a problem that we cannot ignore. Therefore, in this paper, we propose an improved bat algorithm with dynamic inertia weight and time factor. The dynamic inertia weight we used in this paper control every bats' velocity dynamically, at the same time, time factor with weight adjustment for each generation of bats is added to avoid falling into the local optimum. Simulation results indicate that the improved bat algorithm shows excellent performance in solving multidimensional continuous space optimization problems. This paper applies the improved BA to solve Capacitated VRP (CVRP, abbreviation as VRP) [15], and demonstrates that the improved bat algorithm has achieved very good performance in solving discrete problems.

2. VRP Mathematical Model

Vehicle routing problems with capacity limitation are also referred to as standard vehicle routing problems, which means a maximum load capacity limit for each service vehicle. It is the most basic vehicle routing problem-- other variants of the vehicle routing problems are based on this. The general description of VRP is: Given a central warehouse and sub-warehouse, the location coordinates of each warehouse point and cargo demand $g_i (i = 1, 2, \dots, L)$ is known, the vehicle K loading capacity $q_k (k = 1, 2, \dots, K)$ is certain. Each car starts from the central warehouse to complete the distribution of the sub-warehouse and must return to the central warehouse after the task finished. If each sub-warehouse is accessed only once, the sum of the sub-warehouse visits required by each vehicle cannot exceed the vehicle loading capacity, and all sub-warehouse distributions need to be completed with minimum total cost.

According to the above description, the mathematical model of VRP can be established. The central warehouse number is 0, sub-warehouse number is $1, 2, \dots, L$, the demand for goods is $g_i (i = 1, 2, \dots, L)$, vehicle K loading capacity is $q_k (k = 1, 2, \dots, K)$, the definition of variables are as follows:

$$y_{ki} = \begin{cases} 1, & \text{sub-warehouse } i \text{ task by the vehicle } k \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & \text{vehicle } k \text{ from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$$

c_{ij} means transport costs that from point i to point j , actually its meaning can be distance, cost, time, and so on. In this paper, it represents the distance. The mathematical model of VRP is as follows: $X = (x_{ijk}) \in S$

$$\text{Min } Z = \sum_i \sum_j \sum_k c_{ij} x_{ijk}$$

$$\text{s.t. } \sum_i g_i y_{ki} \leq q_k \quad \forall k \quad (1)$$

$$\sum_k y_{ki} = 1, \quad i = 1, 2, \dots, L \quad (2)$$

$$\sum_i x_{ijk} = y_{kj}, \quad j = 0, 1, \dots, L; \quad \forall k \quad (3)$$

$$\sum_i x_{ijk} = y_{ki}, \quad i = 0, 1, \dots, L; \quad \forall k \quad (4)$$

$$X = (x_{ijk}) \in S \quad (5)$$

$$x_{ijk} = 0 \text{ or } 1, \quad i, j = 0, 1, \dots, L; \quad \forall k \quad (6)$$

$$y_{ki} = 0 \text{ or } 1, \quad i = 0, 1, \dots, L; \quad \forall k \quad (7)$$

In this model, the Equation (1) limits the carrying capacity of the vehicle. The Equation (2) ensures that each vehicle can be accessed by each customer only once. The Equation (3)-(5) form a loop. And Equation (6)-(7) is the defined variables. These Equations ensure that each sub-warehouse can get service, and the demand of each point can be completed by a car, while the total demand on each path does not exceed the load capacity of the vehicle. The objective function is fulfilled under such conditions - minimizing the sum of all vehicle travel paths.

As a NP-hard problem, VRP is difficult to solve. Since the issue was put forward, a considerable number of experts and scholars have conducted a great deal of research and laid the foundation for further research in the future.

3. BA and Improved BA

3.1. BA

The bat algorithm is an iterative-based optimization algorithm, similar to most meta-heuristic optimization algorithms. The bionic principle is that the bats are mapped to points in the search space, the search and optimization process is simulated as bats individual searching for prey and moving process. The objective function of the problem is measured as the pros and cons of the bat, the individual survival of the fittest process is simulated as the iterative process which is using better feasible solution to replace poor feasible solution in the optimization process.

It is worth pointing out that the bat algorithm is not just a meta-heuristic algorithm. Compared with the existing meta-heuristic algorithm, it has two advantages: adaptive frequency adjustment and dynamic control of the detection in the global optimization and local optimization free switching. It simulates bat behavior using frequency-based tuning and pulse-rate-of-change, which gives the bat algorithm good convergence and is easier to implement than other algorithms. Bat algorithm uses dynamic search strategy. Like bats in prey on the automatic alignment, pulse emission rate and loudness changes automatically control the search behavior. In fact, when approaching to optimal, switching from local optimization to global optimization automatically can be achieved, therefore, the algorithm can be very effective in practice application problem.

In order to imitate this echolocation system of the micro-bats, Yang (2010) [3] have consider three appropriate and idealized rules as follows:

(1) All bats can use echolocation system to determine the distance, and can distinguish prey or obstacle in some way we cannot imagine;

(2) Bats fly randomly with velocity v_i at position x_i with a frequency f , wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength and frequency of their emitted pulses and adjust the rate $r \in [0,1]$ of pulse emission depending on the proximity of target;

(3) Although the loudness can vary in many ways, we assume that the loudness varies from a large A_0 to a minimum constant value A_{\min} .

The optimization process of bats in BA can be described as follows:

Every bat is randomly distributed in the search space, their location is $x_i (i = 1, 2, \dots, N)$. Bats can emit ultrasonic pulses with frequency f_i , loudness A_0 to search prey. Once confirming the target, they will fly to prey at velocity v_i , and then adjust the velocity, pulse rate and loudness based on the distance from target. According to the process, Yang designed the standard BA's mathematical model, let t for iteration of the algorithm, i for each bat, f_i for frequency that the bat i emitting, v_i for velocity, x_i for the location of bats. Therefore, the frequency, velocity and location updating Equation (8)-(10) is as follows:

$$f_i = (f_{\max} - f_{\min})\beta \quad (8)$$

$$v_i' = v_i^{t-1} + (x_i^t - x_*)f_i \quad (9)$$

$$x_i' = x_i^{t-1} + v_i' \quad (10)$$

In Equation (8), β is a randomly generated number within the interval $[0,1]$. In general, the frequency f in a range $[f_{\min}, f_{\max}]$ corresponds to a range of wavelengths $[\lambda_{\min}, \lambda_{\max}]$. For example, a frequency range of $[20\text{kHz}, 500\text{kHz}]$ corresponds to a range of wavelengths from 0.7mm to 1.7mm (Yang2010).

Actually, we can use any wavelength depending on different problems. The bats can adjust the range with the changing of frequency. In Equation (9), x_* denotes the current global optimal solution gradually, they take local search strategy, using local solution updating Equation (11) as follows:

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A^t \quad (11)$$

x_{old} denotes the solution of the current optimal solution, and ε denotes a random number in the range $[-1,1]$, while A^t is the average loudness of all the bats at time t . Furthermore, with the iterations proceeding, the loudness A_i and the rate r_i of pulse emission must be updated based on these Equations (12) and (13).

$$A_i^{t+1} = \alpha A_i^t \quad (12)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (13)$$

Where α and r are constants, for any $0 < \alpha < 1$ and $\gamma > 0$, we can see that the loudness, A_i , decreases and the pulse rate, r_i , increases at the same time when the bats are near the prey step by step. In the simplicity case, the emission pulse rate, r_i , and loudness, A_i , are often randomly chosen at the first step, and the parameter α and r set to 0.9 generally.

Above all, we can summarize BA's two main strategies—exploration and exploitation. The former means global search strategy. When bats search for prey, its sound frequency is lower and the wavelength is longer and louder. The latter means local search strategy. When bats detect prey, its sound frequency is higher and the wavelength is shorter and quieter. This process will be repeated until the termination criteria met. The standard BA pseudo code is as follows.

Bat algorithm pseudo code

Input: Initialize the bat population x_i , f_i , and v_i , ($i=1,2,\dots,n$)

1. **While** ($t < T_{\max}$);
 2. Update velocities and locations;
 3. **If** ($\text{rand} > r_i$);
 - Select a current best solution randomly among the best solutions and generate a local solution around the selected best solution;
 5. **End if**;
 6. Generate a new solution randomly;
 7. **If** ($\text{rand} < A_i$ and $f(x_i) < f(x_*)$);
 8. $f(x_*) = f(x_i)$;
 9. Increase r_i and reduce A_i ;
 10. **End if**;
 11. Rank the bats and find the current best x_* ;
 12. **End while**.
-

3.2. Improved BA

Inspired by PSO's dynamic inertia weight, this paper chooses a dynamic inertia weight based on Gaussian Normal

Distribution [15-16]. Every generation, bats start searching from a random position with random velocity. With such a dynamic velocity, bats can be distributed throughout the search space immediately at a preliminary stage, which increases the opportunity to jump out of the local optimal solution bats fall into. According to this analysis, the modified velocity Equation (14) and dynamic inertia velocity Equation (15) is as follows:

$$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min})rand() - \sigma \cdot normrnd(0,1) \quad (14)$$

$$v_i^t = \omega \cdot v_i^{t-1} + (x_i^{t-1} - x_s)f_i \quad (15)$$

Where ω_{\max} and ω_{\min} are the maximum ω value and minimum ω value, $rand()$ is a random number subjected to a uniform distribution in a range [0,1]. In this paper, σ denotes the inertia deviation factor, in the range [0.1,0.9], and is a random number subjected to a uniform distribution. In this Equation (14), using $\sigma \cdot normrnd()$ control deviation degree of inertia weight ω . The $normrnd()$ is a random number subjected to Gaussian normal distribution in a range (0,1). In short, such inertia weight not only makes the velocity of every generation of bats have randomness, but also have rationality. This enhances the diversity of bats in every generation and improves the BA's convergence capability.

In physics, only two physical quantities with the same dimensions can be calculated directly. In the standard BA however, the position and velocity are directly added. Therefore, there is an implied time factor with a fixed value constant 1 in this Equation. This is the main reason why the bats vibrate near the optimal solution. So, we can change bat flight time randomly based on the iteration process to improve bat search ability.

According to the analysis above, this paper designs the time factor $t = 0.1 + \omega$, which is changed with the inertia weight change. In the improved bat algorithm, the bats can choose appropriate inertia weight and time factor self-adaptively based on the bat search feature during the iteration process to find the global optimal solution easily. The modified solution updating Equation is as follows-Equation (16):

$$x_i^t = x_i^{t-1} + (0.1 + \omega)v_i^t \quad (16)$$

4. Improved BA for VRP

4.1. The Basic Idea

First of all, the bat algorithm finds the optimal solution in an N-dimensional Hilbert space, and VRP is a typical discrete system optimization problem. The solution is obtained in the form of a sequence satisfying the constraints and the objective function. Using fast sort to map the points in N-dimensional Hilbert space into a solution vector, we can obtain the objective function value of this point by evaluating the cost of the sequence. In this process, each point in the search space corresponds to each bat in the bat algorithm, and the objective function value of each point is the fitness value of each bat.

4.2. Solution Code

In VRP, the central warehouse and sub-warehouses are discrete points, so the solution of real coding is adopted to convert the VRP into a quasi-continuous problem. In this way, the coding method is directly effective. In particular, the existing updating rules of the algorithm can be utilized directly in the running process and the algorithm can give full play to the advantages. The operation dimension of the algorithm is consistent with the problem solving dimension. Suppose there are L sub-warehouses and K cars, then the dimension of VRP solution is $d = L + K - 1$. Each individual bat and velocity vector corresponds to one $(L + K - 1)$ dimension vector. In fact, the resulting path has $(L + K + 1)$ elements, but in this program, code does not need to consider the first and the last two 0s in advance. They only need to add the first and last two 0s when the total path is obtained. This not only has no effect on the calculation of the fitness value of each bat and the search of the optimal solution, but also can reduce the solution dimension and improve the search speed appropriately.

For example, the central warehouse code is 0, sub-warehouse code is $1, 2, \dots, n$, and there are K vehicles involved in distribution. According to the mathematical model described above, if there are 10 sub-warehouses and 3 vehicles involved in distribution, you can get the form of the solution shown below (Figure 1).

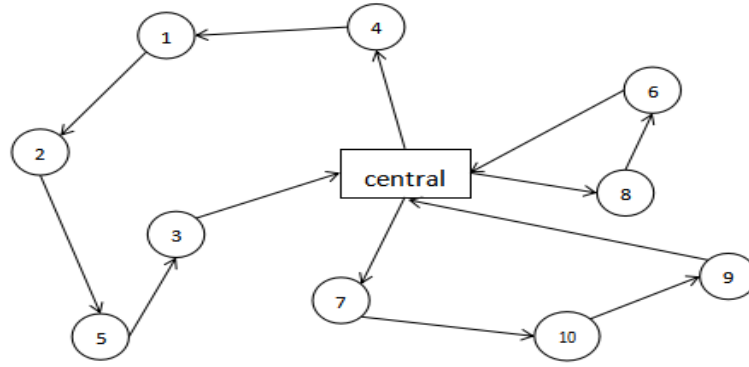


Figure 1. Paths

Sub-path: Vehicle 1 Path: 0-4-1-2-5-3-0.

Vehicle 2 path: 0-7-10-9-0.

Vehicle 3 path: 0-8-6-0.

The total path: 0-4-1-2-5-3-0-7-10-9-0-8-6-0.

4.3. Steps

Step 1 Randomly initialize the bat population individual number N , frequency's maximum value f_{\max} minimum value f_{\min} , velocity v_i , location x_i , loudness A_i^l and pulse rate r_i .

Step 2 Use fast sort to map the individual bat into a solution vector.

Step 3 Evaluate the feasibility of each solution vector, if feasible, switch to step 4, if not, switch to step 2.

Step 4 Based on fitness function $f(x)$, $x = (x_1, x_2, \dots, x_N)^T$, find every bat fitness value and current optimal solution, record the bat current optimal location as x_* .

Step 5 Update velocity and location using (8), (14), (15).

Step 6 Generate a random number, if $rand > r_i$, generate x_{new} around the selected best solution using (11).

Step 7 Generate a random number, if $rand > A_i^l$ and $f(x_i) < f(x_*)$, record x_{new} as x_{best} , then update loudness A_i^l and pulse rate r_i using (12), (13).

Step 8 Evaluate fitness value, select best solution x_* in this generation.

Step 9 Repeat the iteration process of step2 to step6, until finding global optimal solution or satisfying the maximum iteration number.

Step 10 Output global optimal solution.

5. Experiment

5.1. Example 1

For one central warehouse and eight sub-warehouses distribution system, the demand for each sub-warehouse is $d = [1 \ 2 \ 1 \ 2 \ 1 \ 4 \ 2 \ 2]$. The delivery capacity for per car is 8 (in t). The known center warehouse and sub-warehouse distance is shown as below (Table 1). This requires reasonable arrangements for driving routes, which means the shortest distance. The required number of vehicles is calculated according to the Equation given in [17].

$$m = \left\lceil \frac{\sum g_i}{\alpha q} \right\rceil + 1 \quad (16)$$

g_i is for each sub-warehouse cargo demand, q is for the load, α is a random number between (0,1), and $\lceil \cdot \rceil$ is for the rounding function. In Example 1, the number of vehicles is 2.

Table 1. Distance and demand

	0	1	2	3	4	5	6	7	8
0	0	4	6	7.5	9	20	10	16	8
1	4	0	6.5	4	10	5	7.5	11	10
2	6	6.5	0	7.5	10	10	7.5	7.5	7.5
3	7.5	4	7.5	0	10	5	9	9	15
4	9	10	10	10	0	10	7.5	7.5	10
5	20	5	10	5	10	0	7	9	7.5
6	10	7.5	7.5	9	7.5	7	0	7	10
7	16	11	7.5	9	7.5	9	7	0	10
8	8	10	7.5	15	10	7.5	10	10	0
d		1	2	1	2	1	4	2	2

We used the improved bat algorithm proposed in this paper to solve the problems shown above. Each individual bat is mapped into a 9-dimensional (8+2-1) solution vector to represent the vehicle routing scheme. In this experiment, we have run the improved bat algorithm 20 times, and the parameters' setting is as followed: $n = 180$, $A = 4.5$, $r = 4.75$, $t = 600$, $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$. The solution result statistics is as the following Table 2.

Table 2. Solution for 20 times

Algorithm	Times									
	1	2	3	4	5	6	7	8	9	10
Improved BA	67.50	69.00	67.50	69.00	67.50	69.00	67.50	70.00	69.00	69.00
	11	12	13	14	15	16	17	18	19	20
Improved BA	67.50	70.00	67.50	69.00	69.50	70.00	69.00	67.50	67.50	69.00

Among them, the optimal solution is reached 8 times, the average value of solution is 68.575, and the optimal path is: $0 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 0 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 8 \rightarrow 2 \rightarrow 0$. The total distance is 67.5.

At the same time, it can be seen more intuitively through the convergence curve in Figure 2. It is more efficient and faster to solve VRP with the improved BA, which can converge to the optimal solution within 100 generations.

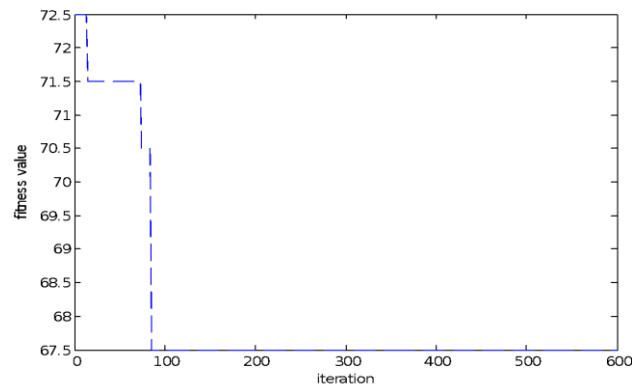


Figure 2. Convergence curve

The reference [18] has studied example1 by using standard genetic algorithm and sub-population genetic algorithm. Using the two different GA, each run 20 times, the results are as the following Table 3.

Table 3. Solution by standard GA and double-population GA for 10 times

Algorithm	Times									
	1	2	3	4	5	6	7	8	9	10
standard GA	74.00	75.00	71.50	72.00	73.50	75.00	73.00	72.50	75.50	73.50
double-population GA	70.00	69.50	67.50	71.00	69.00	70.50	72.00	67.50	71.50	69.00
	11	12	13	14	15	16	17	18	19	20
standard GA	72.00	69.00	73.00	75.50	72.00	73.00	75.00	73.50	71.50	75.00
double-population GA	67.50	69.00	71.00	70.00	67.50	70.50	69.00	69.50	71.00	69.00

It is known that the optimal solution of this problem is 67.5, the standard genetic algorithm reaches 0 times, the double population genetic algorithm reaches 4 times, and the improved bat algorithm reaches the known optimal solution 8 times. Thus, the improved bat algorithm can be used to solve the vehicle routing problem, and has good performance.

5.2. Example 2

We use the internationally VRP database benchmark test cases, the example 2(P-n19-k2) is downloaded from <http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html?/algorithms/clustRout.html/>.

P-n19-k2 has 19 sub-warehouses, and coordinate and demand as shown in the following Table 4.

Table 4. P-n19-k2 coordinate and demand

	Coordinate	Demand		Coordinate	Demand
0	(30,40)	0	10	(42,57)	8
1	(37,52)	19	11	(27,68)	7
2	(49,43)	30	12	(43,67)	14
3	(52,64)	16	13	(58,27)	19
4	(31,62)	23	14	(37,69)	11
5	(52,33)	11	15	(61,33)	26
6	(42,41)	31	16	(62,63)	17
7	(52,41)	15	17	(63,69)	6
8	(57,58)	28	18	(45,35)	15
9	(62,42)	14			

1 is for the central warehouse, (1,...,18) is for sub-warehouse, each car capacity is set to 160, the remaining parameters set to (). In the improved bat algorithm, running 10 times, the results shown in the following Table 5.

Table 5. Solution of P-n19-k2 for 10 times and the average value, standard deviation

	1	2	3	4	5	6	7	8	9	10	AV	Std
P-n19-k2	288	265	266	258	282	265	272	275	274	284	272.9	2.83

The current best solution to this problem is known 212, The optimal path is as follows:

Vehicle 1 Path: 0-4-11-14-12-3-17-16-8-6-0.

Vehicle 2 path: 0-18-5-13-15-9-7-2-10-1-0.

It can be seen from the above results that the improved bat algorithm can be very close to the optimal solution. The solution efficiency and solution accuracy are higher than the standard bat algorithm.

6. Conclusion

The vehicle routing problem is an important part of the logistics distribution system, a reasonable way to determine the delivery path is very important to improve service quality. In this paper, a new real number coding scheme is designed, which transforms the discrete VRP into a quasi-continuous problem and is combined with the improved bat algorithm to solve the vehicle routing problem. The experiments show that the improved bat algorithm has its advantages, such as less

parameters, easy implementation and so on, which can effectively solve VRP. However, the research in this paper is not perfect. It has been found in a large number of case experiments that the number of algorithm population and population mutation rate also have a great influence on the solution. In the future, research can continue based on this research direction.

Acknowledgement

This work is supported by the Science & Technology Program of Henan Province, China (Grant No. 182102310886 and 162102110109). The authors would like to thank the anonymous reviewers and the editor for their helpful comments.

References

1. G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," *Management Science*, Vol. 6, No. 1, pp. 80-91, 1959
2. N. Christofides, A. Mingozzi, and P. Toth, "Exact Algorithms for the Vehicle Routing Problem based on Spanning Tree and Shortest Path Relaxations," *Mathematical Programming*, Vol. 20, No. 1, pp. 255-282, 1981
3. F. Glover and M. Laguna, "Tabu Search," *General Information*, Vol. 106, No. 2, pp. 221-25, 2006
4. D. E. Goldberg, "Genetic Algorithm in Search Optimization and Machine Learning," *Addison Wesley* xiii, Vol. 7, pp. 2104-2116, 1989
5. L. M. Gambardella, E. D. Taillard, and G. Agazzi, "A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows," 1999
6. J. Brandão, "A Tabu Search Algorithm for the Open Vehicle Routing Problem," *European Journal of Operational Research*, Vol. 157, No. 3, pp. 552-564, 2004
7. E. Choi and D. W. Tcha, "A Column Generation Approach to the Heterogeneous Fleet Vehicle Routing Problem," *Computers & Operations Research*, Vol. 34, No. 7, pp. 2080-2095, 2007
8. G. Nagy and S. Salhi, "Heuristic Algorithms for Single and Multiple Depot Vehicle Routing Problems with Pickups and Deliveries," *European Journal of Operational Research*, Vol. 162, No. 1, pp. 126-141, 2005
9. X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," *Computer Knowledge & Technology*, Vol. 284, pp. 65-74, 2010
10. X. S. Yang, "Bat Algorithm for Multi-Objective Optimization," *International Journal of Bio-Inspired Computation*, Vol. 3, No. 5, pp. 267-274, 2012
11. C. Karri and U. Jena, "Fast Vector Quantization using a Bat algorithm for Image Compression," *Engineering Science & Technology an International Journal*, Vol. 19, No. 2, pp. 769-781, 2016
12. A. Chakri, R. Khelif, and M. Benouaret, X. S. Yang, "New Directional Bat Algorithm for Continuous Optimization Problems," *Expert Systems with Applications*, Vol. 69, pp. 159-175, 2017
13. P. Musikapun and P. Pongcharoen, "Solving Multi-Stage Multi-Machine Multi-Product Scheduling Problem using Bat Algorithm," *International Proceedings of Economics Development & Research*, Vol. 35, 2012
14. G. G. Wang, H. C. E. Chu, and S. Mirjalili, "Three-Dimensional Path Planning for UCAV using an Improved Bat Algorithm," *Aerospace Science & Technology*, Vol. 49, pp. 231-238, 2016
15. P. Augerat, J. M. Belenguer, E. Benaventb, A. Corberan, and D. Naddef, "Separating Capacity Constraints in the CVRP using Tabu Search," *European Journal of Operational Research*, Vol. 106, No. 2-3, pp. 546-557, 1998
16. R. C. Eberhart, Y. Shi, "Tracking and Optimizing Dynamic Systems with Particle Swarms," *Evolutionary Computation*, in *Proceedings of the 2001 Congress on IEEE*, Vol. 1, pp. 94-100, 2001
17. Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," *Advances in Natural Computation*, pp. 439-439, Springer Berlin Heidelberg, 1998
18. J. Li, "Genetic Algorithm for Vehicle Scheduling Problem with Non-Full Load," *SYSTEMS ENGINE-THEORY METHODOLOGY APPLICATION*, 2000

Yu Li received her PhD from University of Shanghai for Science and Technology. She is a professor of Institute of Management Science and Engineering, and Business School, Henan University. Her research interests include intelligence algorithms, electronic commerce, etc.

Qian Guo is a master degree candidate of Business School, Henan University. Her research interest is intelligence algorithm.

Jingsen Liu received his PhD from Northwestern Poly-technical University. He is a professor of Institute of Intelligent Network system, and College of Software, Henan University. His research interests include intelligence algorithm and network information security, etc.