Research on the Vehicle Routing Problem with Time Windows Using Firefly Algorithm

Fengshan Pan[1], Chunming Ye[1], Kefeng Wang[*2], Jiangbo Cao[1]
1.Business School , University of Shanghai for Science and Technology, Shanghai, China. Email: yechm6464@163.com
2.School of Energy Science and Engineering, Henan Polytechnic University, Jiaozuo, China. Email: kfhere@126.com

***Abstract*—As a key factor of logistics distribution, vehicle routing problem (VRP) is a hot research topic in management and computation science. In this paper the principle and algorithm process of firefly algorithm are introduced in detail. Then the algorithm process and solving steps are designed for vehicle routing problem with time windows (VRPTW), including the coding and design of disturbance mechanism of elicit fireflies. In the end, the testing examples from benchmark and other literatures are conducted with good outputs, which prove the validity of the firefly algorithm for VRP.**

***Index Terms*—vehicle routing problem; time windows; firefly algorithm; optimization**

I. Introduction

Logistics distribution plays an important role in the modern logistics management. The vehicle routing problem (VRP), as a key issue in logistics distribution, has become a hot topic for experts in management and computation research fields during the last few decades. As we all know, reasonable scheduling of vehicles and vehicle routes can reduce distribution costs, thereby reducing the whole logistics costs. Since the problems are very common in the real world, such as the supermarket distribution, freight transportation and so on, therefore, VRP is a very practical and interesting issue.

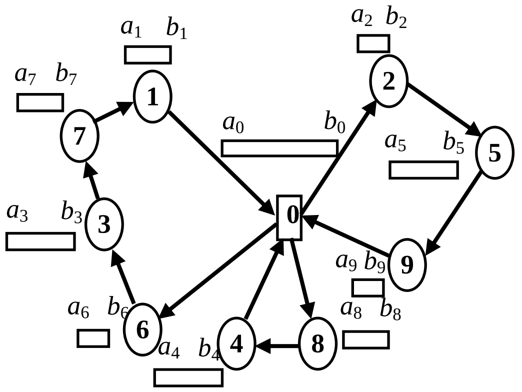

Figure 1. A Figure Illustrating VRPTW

Since Yang [1, 2] proposed the firefly algorithm (FA) for multimodal function optimization, FA has been applied to other various optimization problems over the past four years. Yang has also compared FA with the fully-developed particle swarm optimization algorithm (PSO) and genetic algorithm (GA)[3,4].

Few literatures about FA can be found, but more and more scholars abroad begin to pay attention to the algorithm. Some of them have made several improvements, which enable the algorithm to solve not only function optimization problems but also practical problems. Sayadi et al.[5] suggested the discrete firefly algorithm (DFA）and applied it to solving flow-shop problems which belong to production scheduling problems. Broersma[6] put forward a multi-objective firefly algorithm, a revised FA, to solve the load distribution problem in economics. Rampriya et al [7] presented a kind of Lagrange FA algorithm, and applied it to the combination optimization of electric unit in electric power system. In addition, Jati GK and Suyanto[8] used DFA to solve TSP problem; Khadwilard et al.[9] used FA to solve job-shop problem in production scheduling. They all have obtained good results using firefly algorithm. What is worth mentioning here that several domestic literatures about FA have appeared. Liu and Ye[10, 11] applied FA to the permutation flow shop scheduling problems successfully and yielded favorable outcome.

The literatures about FA have showed it is a very promising algorithm. Though great progress has been made on the algorithm in theory and application since it was proposed four years ago, further studies should still be developed on the algorithm, especially in the convergence analysis and applications to other fields.

II. Vehicle Routing Problem with Time Windows

There're different kinds of VRP problems according to the different constraints. The more variables and restrictions there are, the more difficult to solve the problem. CVRP is on the basis of the VRP with a constraint of the vehicle capacity constraints, which is described as follows [12]: a certain number of clients whose locations and delivery demands are presented in advance; a fleet of identical vehicles with fixed capacity leaves from the distribution center and returns after finishing all the delivery tasks. The objective is to minimize the total costs. Several assumptions are provided here: each client will be visited by one vehicle for only once and the total demands of visited clients should be no more than the vehicle capacity. The vehicle routing problem with service time limitation at every

*Corresponding author.

doi:10.4304/jcp.8.9.2256-2261

client and the distribution center is called VRPTW[12]（vehicle routing problem with time windows, abbreviated as VRPTW）, developing from the CVRP with consideration of time windows. The time window refers to a time range within which one client is started to be served. If a vehicle arrives earlier than the start point of the client's time window, it has to wait. The objective of VRPTW is to complete all the delivery tasks of all clients and minimize the total costs of distribution. Figure 1 illustrates the VRPTW problem.

Many research results about VRPTW have been made since VRP was proposed, which are mainly about the solving methods such as: the accurate branch-and-bound method, heuristic method based on experience, tabu search method, genetic algorithm, particle swarm optimization algorithm, ant colony algorithm [13, 14, 15] and so on, every of which adapts to solving some kinds of problems. In this paper a kind of firefly algorithm is designed to solve VRPTW and it performs very well by many computational tests.

III. MODEL DESCRIPTION OF VRPTW

Mathematical model of VRPTW is described as follows: we assumes there're $K(k=1,2,\cdots,K)$ vehicles at the distribution center and $L(i=1,2,\cdots,L)$ customers, and $i=0$ represents the distribution center. The vehicle capacity is q; the demand of every client node is $g_i(i=1,2,\cdots,L)$; the delivery costs from client node i to client node j is c_{ij} (it could be distance or fees etc); time window at client i is denoted as $[e_i, l_i]$; let se_i to be the time when the service at client node i begins, sl_i the time when the service at client node i ends, and s_i the total service time at client node i, so $s_i = sl_i - se_i$; t_{ij} is the travel time from client node i to client node j. Here we simply take $t_{ij} = c_{ij}$ for convenience. Besides, two variables are defined:

$$y_{ik} = \begin{cases} 1 & \text{the demand of client } i \text{ completed by vehicle } k \\ 0 & \text{others} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{vehicle } k \text{ travel directly from client } i \text{ to client } j \\ 0 & \text{others} \end{cases}$$

So the mathematical model of VRPTW is presented as follows:

$$MinZ = \sum_{k=1}^{K}\sum_{i=1}^{L}\sum_{j=1}^{L} c_{ij}x_{ijk}$$

s.t.
$$\sum_{i=1}^{L} g_i y_{ik} \le q, \qquad \forall k \tag{1}$$

$$\sum_{k=1}^{K} y_{ik} = 1, \qquad \forall i \tag{2}$$

$$\sum_{i=1}^{L} x_{ijk} = y_{ik}, \qquad \forall k, j \tag{3}$$

$$\sum_{j=1}^{L} x_{ijk} = y_{ik}, \qquad \forall k, i \tag{4}$$

$$\sum_{i,j}\sum_{\in S\times S} x_{ijk} \le |S| - 1, S \subset \{1, 2...L\}, \forall k \tag{5}$$

$$sl_i + t_{ij} \le se_j, \qquad \forall k, i, j \tag{6}$$

$$e_i \le se_i \le s_i, \qquad \forall i \tag{7}$$

$$x_{ijk}, y_{ik} \in (0,1). \qquad \forall k, i, j$$

Where $|S|$ is the total nodes number in set S. Constraint (1) limits the vehicle's load. Constraint (2) makes sure each client is visited only once by each vehicle. Constraints (3)、(4)、(5) are designed to ensure a complete circuit. Constraint (6) gives the premise for the two adjacent tasks on one route. Constraint (7) is about time window restriction.

VRPTW is the most studied type of vehicle routing problem since its model is more close to the real life and describes the real problem better. VRPTW can be further divided into two types: VRP with hard time windows and VRP with soft time Windows. Generally speaking, vehicle routing problem with time windows, without specific explanation, refers to VRP with hard time windows.

IV. SOLVING VRPTW BY FA

A. Coding Design

VRPTW is developed from CVRP with the consideration of time windows. Therefore we can adopt the same real-coded schema for VRPTW as for CVRP. In VRPTW with L client nodes and K vehicles, fireflies' positions are represented by vectors with $L+K-1$ dimensions so as to match with the final distributing solution.

This paper adopts the real-coded schema, introduced by Wu[15] for PSO algorithm in VRP, for VRPTW coding. We mark $1, 2, \ldots, L$ as the client nodes and 0 as the distribution center. Since there are K vehicles for distribution so at most K routes are considered. Each vehicle starts and ends at the distribution center. $K-1$ virtual centers represented by $L+1, L+2, \cdots, L+K-1$ are considered in the coding design to describe the distribution routes better. The problem is coded in such method that each firefly corresponding to a $L+K-1$ dimensional vector , which valued sequence conveys the information of distribution order on the total delivery routes. The firefly's position vector in such a way matches with the final solution.

For instance, assume there are 8 clients, and 3 vehicles, the position vector of firefly x is:

Warehouse number:	1	2	3	4	5	6	7	8	9	10
x:	4.9	7.0	3.5	1.2	1.3	8.0	4.2	8.7	1.9	6.6

For the convenience of calculation, rounding treatment on x is given according to the original order of each dimensional vector-value. The result is showed as follows:

Warehouse number: 1 2 3 4 5 6 7 8 9 10
x: 6 8 4 1 2 9 5 10 3 7

So the position vector of firefly provides a solution of total routes in the following sequence: 0→4→5→9→3→7→1→10→2→6→8→0 , where 9 and 10 are virtual centers.

The distribution route of each vehicle is given here:

Vehicle 1: 0→4→5→0;
Vehicle 2: 0→3→7→1→0;
Vehicle 3: 0→2→6→8→0.

The total routes set of client nodes sequence has $L+K-1$ numbers, beginning with zero. Two zeros are not allowed to be adjacent. The same as the solution in reference [15]. This paper doesn't consider the cases of two adjacent zeros, because there doesn't exist any feasible result at that time and thereby the cases are automatically ruled out during the iterations. The two zeros at the end of the sequence will be removed at first while coding the firefly vector, and added back while decoding the firefly vector into client nodes to calculate the minimum total costs. The coding process is designed in such a way to guarantee a correct optimal solution and a quicker search since the dimension of a firefly vector is reduced.

B. Modifying Infeasible Solution

The position vector of a firefly keeps changing in the search process by following the brightest firefly in the group. The updated vector cannot guarantee different dimensional vector-values after rounded treatment. While the effectiveness of solutions requires the values on position vector of each firefly to be different and covered in the range of coding. Hence only one will be kept among each array of duplicate values, if there are, and other duplicate values will be replaced with unused numbers in the range from 1 to $L+K-1$.

C. Constraints

This paper uses the penalty function method to describe the fixed vehicle capacity constraint in VRPTW, just as in CVRP. The penalty function is formed by adding a large positive number R to the loading capacity of each vehicle. Then the penalty function is added to the objective function, as shown in formula (8).

$$MinZ = \sum_{i=0}^{L}\sum_{j=0}^{L}\sum_{k=1}^{K} c_{ij} x_{ijk} + R \times \sum_{k=1}^{K} \max\left(\sum_{i=1}^{L} g_i y_{ik} - q, 0\right) \qquad (8)$$

D. Basic Principle of FA

Fireflies' attraction to each other depends on two factors: its brightness and attraction. Brightness is determined by the function value of the fireflies' position. The brighter the firefly is, the better its location is. Attraction is associated with brightness. The brighter firefly is, the stronger its attraction is to other fireflies with a lower brightness. If there is no firefly with higher brightness than itself, a firefly just flies randomly. FA achieves the global optimization by fireflies' continuous updating position based on the brightness and attraction.

(1) Distance between fireflies

Distance between firefly i and firefly j are defined by their Cartesian distance, as shown in (9).

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}\ , \qquad (9)$$

Where $x_{i,k}$ is the k th component of firefly i 's position vector, d is the position vector's dimension.

(1) Relative Brightness

Brightness of one firefly at the position of x is represented by I , its relative brightness is computed as follows:

$$I = I_0 \times e^{-\gamma r_{ij}}, \qquad (10)$$

Where I_0 is its initial brightness, namely its brightness at $r=0$. I_0 is associated with the objective function value at $r=0$ and the smaller the function value is, the bigger I_0 will be. γ represents the degree of light attenuation, reflecting the characteristics that fireflies' brightness decreases with distance's increase and the media's absorption.

(2) Degrees of Attraction

Degrees of attraction can be yielded with the following equation:

$$\beta = \beta_0 \times e^{-\gamma r_{ij}^2}, \qquad (11)$$

Where β_0 is the largest degree of attraction, namely the degree of attraction at $r=0$.

(3) Position Update

Position update when firefly i is attracted to the brighter one j , which is given as follows:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(rand - 1/2) \qquad (12)$$

Here, $\beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i)$ is the attraction of firefly j to firefly i . α is the step factor, a random number in the range of $[0,1]$, and $rand$ is a random number uniformly distributed in the range of $[0,1]$.

E. Steps of Firefly Algorithm for VRPTW

Step1: Defining parameters

We set the number of fireflies is m , the largest degree of attraction is β_0 , the degree of light attenuation is γ , the step factor is α , and the maximum number of iterations is $\max Gen$ **.**

Step2: Initializing the population randomly

Take an integer in the range of $[1, L+K-1]$ for each dimension vector-value of firefly's position vector x . Then calculate the objective function value $F_i(0)$ as respective maximum fluorescence brightness I_0 of the fireflies.

Step3: Finding the best firefly

Calculate the relative brightness I and the degree of attraction β according to formula (10) and (11) respectively. Then we can find the brightest firefly, i.e. the most optimal one in the population.

Step4: Updating position and disturbance testing

Update the fireflies' positions based on the formula (12) and modify infeasible solutions. Then disturbance tests are conducted for the M elite fireflies with bigger brightness than others, in order to rule out local optimal solutions.

Disturbance type is designed by mutating a value in a randomly-chosen dimension of the elite fireflies' position. If we use a v as the original value, the new value after mutation v' is calculated with the following formula:

$$\begin{cases} v'(t) = v(t) \pm \delta \cdot \omega(t) \\ \omega(t) = \max Gen - \dfrac{t}{\max Gen} \end{cases} \tag{13}$$

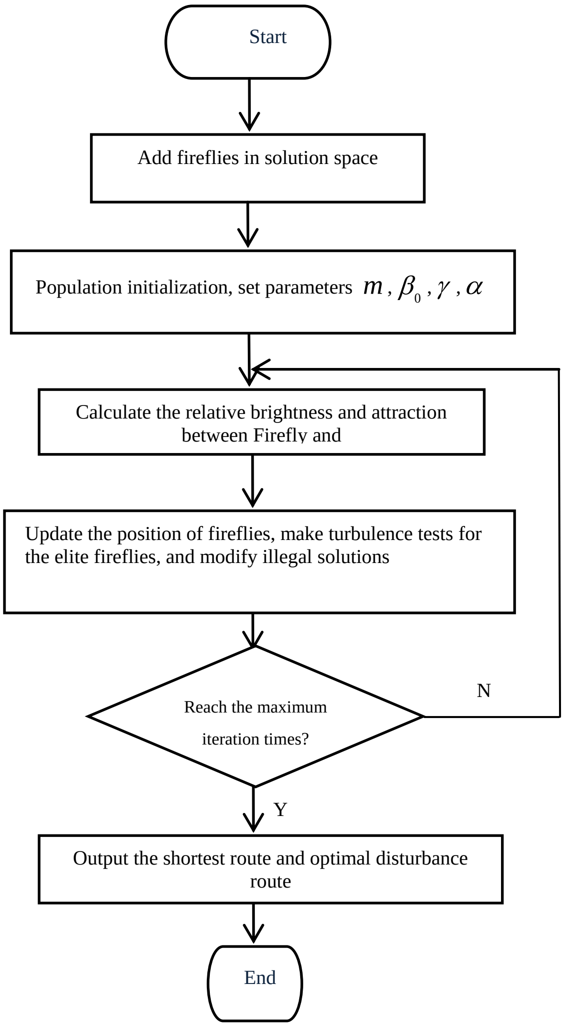

Figure 2. FA Flowchart

where "+" or "-" is decided randomly, δ is a random variable between 0 and 1. $\omega(t)$ is a parameter in the t-th iteration process and it is correlated negatively with the iteration times.

Step5: Termination of the iteration

If the termination condition is met, output the shortest route and the total minimum of distribution routes. if not, turn back to Step3.

Figure 2 is the flowchart illustrating the firefly algorithm process.

V. Experimental Tests

Two examples as the benchmark problems are given in this paper to test the proposed FA algorithm. We adopt a testing example suggested by Wu et al.[16] and testing material by Marius [17] respectively in Example 1 and Example2. Testing results show that the firefly algorithm converges fast to the optimum and yields good solutions.

Example 1：This paper first uses the example in the literature [16] for testing. There are 8 clients (number 1, 2, ... 8) in the example. Demand g_i , service time T_i and time window $[e_i, l_i]$ of each client are known, as shown in Table 1. The distances between nodes are shown in Table 2. Zero represents the distribution center. Vehicle's maximum load is 8 tones.

Computations have been performed to test FA for solving the problem above, using the MatlabR2008 software running under Intel Atom N270 1.6GHz Quad-Core CPU, 1GB Ram, and Windows 7.0 operating system. The main parameters are set as follows:

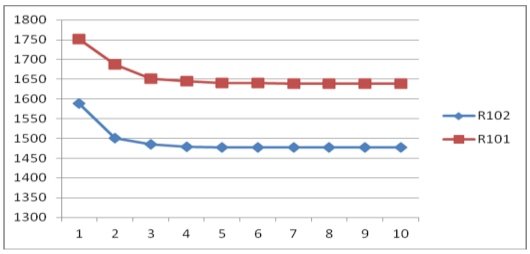

Figure 3. Convergence of the Best Solution Based on FA in R102 and R102

The number of fireflies m : 50

Degree of light attenuation γ： 1.0

Step factor α ： 0.4

Iteration times：300

After the iterations of 300 times, FA converged to the known optimal solution, with total distribution distances of 910 and the distribution routes are as follows.

Vehicle 1: 0→8→5→7→0；

Vehicle 2: 0→3→1→2→0；

Vehicle 3：0→6→4→0 .

Example 2: Marius' six data sets [17] are used as the testing material. Nodes in sets R1 and R2 are randomly located while nodes in sets C1 and C2 are regional aggregated.

Situations in each set, R101 and R102 for instance, have the same nodes' positions, but different time windows. Reference data are also listed here: the optimal solution obtained with an accurate algorithm (absent for some situations as shown in the table with "-") and the best approximate solutions yielded by heuristic algorithms. The optimal solutions by FA in this paper fall somewhere between the exact solutions and the best approximate solutions.

Obviously, for R1 and R2, all results are absolutely (100%) better than the known best approximate solutions. For C1, results in 7 groups（77.7%）are equal to the best approximate solutions, the same to 6 groups（75%）for C2. Overall, our results are all better or equal to the best known approximate solutions.

We provide the convergence speed of the best solutions in R101 and R102 by FA in Figure 3. The vertical axis represents optimal solution value and the horizontal axis represents iteration times (1 unit= iteration of 30 times).

TABLE I.
CLIENTS' DEMANDS AND TIME WINDOW

Client i	1	2	3	4	5	6	7	8
q_i	2	1.5	4.5	3	1.5	4	2.5	3
T_i	1	2	1	3	2	2.5	3	0.8
$[e_i, l_i]$	[1,4]	[4,6]	[1,2]	[4,7]	[3,5.5]	[2,5]	[5,8]	[1.5, 4]

TABLE II.
DISTANCES BETWEEN NODES

i \ j	0	1	2	3	4	5	6	7	8
0	0	40	60	75	90	200	100	160	80
1	40	0	65	40	100	50	75	110	100
2	60	65	0	75	100	100	75	75	75
3	75	40	75	0	100	50	90	90	150
4	90	100	100	100	0	100	75	75	100
5	200	50	100	50	100	0	70	90	75
6	100	75	75	90	75	70	0	70	100
7	160	110	75	90	75	90	70	0	100
8	80	100	75	150	100	75	100	100	0

TABLE III.
THE TEST RESULTS OF SOLOMON R GROUP AND C GROUP

Set of R 1	R101	R102	R103	R104	R105	R106	R107	R108	R109	R110	R111	R112
accurate solutions	1637.7	1466.6	1208.7	971.5	1355.3	1234.6	1064.6	-	1146.9	1068	1048.7	-
best approximate solutions	1645.79	1482.12	1292.68	1007.24	1377.11	1251.98	1104.66	960.88	1194.73	1118.59	1096.72	982.14
Our solutions	1639.2	1476.7	1256.3	994.31	1368.3	1247.4	1078.3	957.56	1169.7	1094.5	1065.2	971.36
Set of R2	R201	R202	R203	R204	R205	R206	R207	R208	R209	R210	R211	
accurate solutions	-	-	-	-	-	-	-	-	-	-	-	
best approximate solutions	1252.37	1191.70	939.54	825.52	994.42	906.14	893.33	726.75	909.16	939.34	892.71	
Our solutions	1241.23	1186.48	938.40	823.87	993.7	905.67	892.59	724.89	908.32	938.8	891.56	
Set of C1	C101	C102	C103	C104	C105	C106	C107	C108	C109			
accurate solutions	827.3	827.3	826.3	822.9	827.3	827.3	827.3	827.3	827.3			
best approximate solutions	828.94	828.94	828.06	824.78	828.94	828.94	828.94	828.94	828.94			
Our solutions	828.94	827.64	827.53	824.87	827.92	827.67	827.85	828.31	828.98			
Set of C2	C201	C202	C203	C204	C205	C206	C207	C208				
accurate solutions	589.1	589.1	588.7	588.1	586.4	586.0	585.8	585.8				
best approximate solutions	591.56	591.56	591.17	590.90	588.88	588.49	588.29	588.32				
Our solutions	591.56	591.56	589.23	589.67	587.21	588.83	587.35	588.39				

VI. Conclusions

Vehicle routing problem is an important problem in logistics distribution. There are many algorithms such as genetic algorithm[18], branch-and-bound algorithm[19] , particle swarm optimization algorithm[20] , artificial bee colony algorithm[21] and so on. FA, as a new type of swarm intelligence algorithm, has enjoyed rapid development since it was introduced. Its application has expanded from function optimization to production scheduling, TSP problem and etc. This paper attempts to apply FA to solve VRPTW, and provides new application possibility of FA. Further researches are needed about VRPTW and VRP with interference in the future. FA convergence analysis is also an important work. Otherwise the algorithm can be extended to address other optimization problems in practice.

Acknowledgements

This paper is funded by the National Natural Science Foundation of China (71271138), Humanities and Social Sciences Planning Fund from Ministry of Education (10YJA630187), Research and Innovation Fund of Shanghai Municipal Education Commission (12ZS133).

References

[1] X.S. Yang, "Nature-Inspired Met heuristic Algorithm ". *Luniver press*, pp. 83-96, 2008.

[2] X.S. Yang, "Firefly algorithms for mult-imodal optimization". *Proc of the 5th International Symposium on Stochastic Algorithms: Foundations and Applications*, pp. 169-178, 2009.

[3] X.S. Yang, "Firefly algorithm, stochastic test functions and design optimisation". *Bio-Inspired Computation*, vol. 2, no. 2, pp. 78-84, 2010.

[4] X.S. Yang, S. DEB, "Eagle strategy using lévy walk and firefly algorithms for stochastic optimization". *Studies in Computational Intelligence*, vol. 284, pp. 101-111, 2010.

[5] M.K. Sayadi, R. Ramezanian, N. Ghaffari-Nasab, "A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems". *Industrial Engineering Computations*, vol. 1, pp. 1-10, 2010.

[6] H. Broersma, "Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem". *International Journal of Combinatorics*, pp. 1-23, 2011.

[7] B. Rampriya, K. Mahadevan, S. Kannan, "Unit commitment in deregulated power system using Lagrangian firefly algorithm". *Proc. of IEEE Int. Conf. on Communication Control and Computing Technologies*, pp. 389-393, 2010.

[8] G.K. Jati, Suyanto. "Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem". *Lecture Notes in Computer Science*, vol. 6943, pp. 393-403, 2011.

[9] A. Khadwilard, S. Chansombat, T. Thepphakorn, P. Thapatsuwan, W. Chainat, P. Pongcharoen, "Application of firefly algorithm and its parameter setting for job shop scheduling". *First Symposius on Hands-On Research and Development*, pp. 89-97, 2011.

[10] C.P. Liu, C.M. Ye, "Novel bioinspired swarm intelligence optimization algorithm: firefly algorithm". *Application Research of Computers*, vol. 2, no. 9, pp. 3295-3297, 2011.

[11] C.P. Liu, C.M. Ye, "Solving Permutation Flow Shop Scheduling Problem by Firefly Algorithm". *Industrial Engineering and Management*, vol. 17, no. 3, pp. 56-59, 2012.

[12] J. Li, Y.H. Guo, "Vehicle optimal scheduling theory and method of logistics distribution". *Beijing: China Fortune Press*, 2001.

[13] X.L. Cui, L. Ma, B.Q. Fan, "Ant searching algorithm for vehicle routing problem". *Journal of Systems Engineering*, vol. 19, no. 4, pp. 418-422, 2004.

[14] B. Wang, X.C. Shang, H.F. Li, "Hybrid simulated annealing algorithm for solving vehicle routing problem". *Computer Engineering and Design*, vol. 30, no. 3, pp. 651-653, 2009.

[15] N. Li, T. Zou, D.B. Sun, "Particle swarm optimization for vehicle routing problem", *Journal of Systems Engineering*, vol. 19, no. 6, pp. 596-600, 2004.

[16] Y. Wu, C.M. Ye, H.M. Ma, M.Y. Xia, "Parallel particle swarm optimization algorithm for vehicle routing problems with time windows". *Computer Engineering and Applications*, vol. 43, no. 14, pp. 223-226, 2007.

[17] M.M. Solomon, J. Desrosiers, "Time Window Constrained Routing and Scheduling Problems: A Survey". *Transportation Science*, vol. 22, no. 1, pp. 1-11, 1988.

[18] R.J. Wang, Y.H. Ru and Q. Long, "Improved Adaptive and Multi-group Parallel Genetic Algorithm Based on Good-point Set". *Journal of Software*, vol.4, no.4, pp. 348-356, 2009.

[19] B.L. Ma, L. Geng, J.B. Yin, L.P. Fan, "An Effective Algorithm for Globally Solving a Class of Linear Fractional Programming Problem". *Journal of Software*, vol. 8, no.1, pp. 118-125, 2013.

[20] Y.J. Hu, G.J. Wang, X.D. Cao, L. Yan, "Application of Particle Swarm Optimization Algorithm based on Classification Strategies to Grid Task Scheduling". *Journal of Software*, vol. 7, no.1, pp. 118-124, 2012.

[21] J. Pan, G.H. Mao, J.X. Dong, "Artificial Bee Colony Algorithm with Local Search for Numerical Optimization". *Journal of Software*, vol. 6, no. 3, pp. 490-497, 2011.