An improved firefly algorithm for traveling salesman problems

WANG Ming-bo[1], FU Qiang[*1,2], TONG Nan[1], LI Mengmeng[1],ZHAO Yiming[1]

[1]College of Science and Technology, Ningbo University, China

No.505,YuXiu Road,Ningbo 315212,China

[2] Faculty of Information Science and Engineering, Ningbo University, Ningbo China

No.808,FengHua Road,Ningbo 315211,China

*corresponding author, e-mail: fuqiang@nbu.edu.cn

KeyWords: fireflies algorithm, Traveling Salesman Problem, mutation mechanism, Elite Strategy, C2Opt operator

Abstract: Firefly algorithm, a new type of swarm intelligence in optimization, has a good performance in combinatorial optimization problem. For Traveling Salesmen Problem (TSP), this process proposes an optimized path approach based on the modified firefly algorithm. In order to avoid a local optimum, an elitist strategy combined with mutation mechanism and the mechanism of fluorescence change are used to help the fireflies group go out of the local optimal area. For problems on intersection of sets, C2Opt operator can be used on the global optimization of the path. Finally, the experimental result would be compared with other classical algorithm to verify the reliability via TSPLIB instance test.

1 Introduction

TSP is a classic problem involving combinatorial optimization with its main objective to find a shortest path that a businessman can have a single trip covering all cities and finally return to the starting point. When the city size, is small, the traditional exhaustion method can be used to effectively search for an optimal path. However, when the city size is too large, “a combinatorial explosion” might occur and would lead exhaustive search to taking too long time to get into trouble. In recent years, inspired from the bionic mechanism, people have introduced a number of swarm intelligence algorithms for solving TSPs. For example: Genetic Algorithm (GA), Ant Colony Optimization (ACO), Glowworm Swarm Optimization (GSO) [1], as well as some new intelligent algorithm: Intelligent Water Drops algorithm (IWDA) [2] , Mosquito Host-Seeking (MHS) , bacterial foraging optimization (DBFO) and water cycle algorithm etc. Meeting the large-scale TSP, the above algorithms will still easily get stuck to a local optimum and hardly meet the requirements of focus on both time and precision.

Probing into this research, people have put forward many improved solutions to the above problems. For example, Ji [3] combines ACO and multi-granularity to resolve TSPs. When city size is relatively large, the algorithm can be effective to improve efficiency and shorten the time of the traditional ACO. However, when the city cluster is not obvious, the tolerance of the optimal solution searched by this algorithm would be comparatively large. Wu [4] indicates the ACO has a low rate of convergence, which might get into a local optimum easily, and he has proposed ACO for a strategy of natural selection, which effectively improves algorithmic accuracy and accelerates the convergence time. Mavrouniotis [5] indicates in terms of TSPs in large-size cities, and put forward an idea of using clustering for path optimization method to cluster, classify and re-combine the large-size cities, reducing the problem complexity and greatly shortening the time that used to be too long due to the large size of the city. Thirachit Saenphon[6] has applied the strategy of inverse gradient search into ACO, not only accelerating the search speed but improving the search accuracy.

Firefly algorithm (FA), a very popular swarm intelligence algorithm in recent years, was proposed by Cambridge scholar Yang in 2008. The essence of the algorithm is a algorithm of random heuristic search which simulate the search and optimization process of fireflies as the

© 2016. The authors - Published by Atlantis Press

process of individual firefly attraction and location update. With a simple structure of the algorithm and fewer parameters to be adjusted as well as a good ability of optimal search, it has drawn the attention of scholars home and abroad, and has been widely used in many areas such as combinatorial problems [7], automatic control [8], cluster [9] and software testing [10] with good results.

Domestic and oversea scholars have reached some achievements in researching how Firefly Algorithm solves TSP. However, the algorithm gets easily into a local optimum due to the big difference in fluorescence on the later stage. Therefore, this dissertation introduces the mechanism of fluorescence change and puts forward the improved solution of combining elite strategies and mutation mechanism to effectively improve the algorithmic optimality. Meanwhile, this article uses C2Opt operator to solve the problems on intersection of sets, improving the optimal accuracy and convergence rate of this algorithm in a further step.

2 A Brief Introduction of Firefly Algorithm

Firefly Algorithm, as a random optimization algorithm, has simulated the luminescent behavior of fireflies to search for food in Nature, putting the search and optimization process of fireflies as the process of individual firefly attraction and location update.

In the algorithmic optimization process, fluorescent brightness and fireflies attraction are two key factors in the evolution of firefly locations. Fluorescent brightness is related to the location of individual firefly as the better location, the higher brightness. The individual firefly with a bad location would move in a direction that depends on the firefly attraction with a good location as the larger attraction of the firefly, the easier it is to attract other fireflies closing to its location. When two types of fluorescent brightness are the same, the individual firefly will be randomly oscillating. With the constant update on individual firefly fluorescent brightness and attraction, firefly swarm will eventually come together and achieve the algorithmic optimization effect. The algorithm is described from a mathematical perspective as below:

Supposed the amount of firefly is m and the dimension for search space is n-dimension, then the initial position of firefly i can be expressed as $x_i=(x_{i1},x_{i2},x_{i3},\ldots,x_{iN})$ in n-dimensional space.

Definition 1: fluorescent brightness of Firefly i is represented as

$$I_{oi}=f(x_i) \qquad (1)$$

$f(x_i)$ represents the location fitness of Firefly i.

Definition 2: The relative fluorescent brightness of Firefly i

$$I_i=I_{oi}\cdot exp(-\gamma\cdot r_{ij}) \qquad (2)$$

Among which: γ represents intensity absorption factor with its value associated with the area, generally set as a constant. $r_{ij}=|x_i-x_j|$, represents the distance between Firefly i and Firefly j.

Definition 3: Firefly's attraction

$$\beta=\beta_0\cdot exp(-\gamma\cdot r^2{}_{ij}) \qquad (3)$$

Among which: β_0 is the factor of maximum attraction. Equation (3) describes the firefly with higher brightness has the attraction that will be decreased with the increasing distance and reducing communication media.

Definition 4: The updated location of Firefly i attracted by Firefly j

$$x_i(t+1)=x_i(t)+\beta\cdot(x_j(t)-x_i(t))+\alpha\cdot(rand-1/2) \qquad (4)$$

Among which: α is step length factor, *rand* is a random factor distributed uniformly on [0,1]. In order to increase the scope of search space and prevent it from being convergent early, a disturbing item of $\alpha\cdot(rand-1/2)$ is added.

In the swarm, Firefly i is always searching for the fireflies that has higher fluorescent brightness than itself. When Firefly j meet the requirements, it will update its own location by using equation (4); when the two types of fluorescent brightness are the same, it will be randomly oscillating; after the firefly update its locations for multiple times, all fireflies will gather together eventually in the brightest location, completing the optimal process.

3 Design of the Improved Firefly Algorithm for solving TSPs

Basic Firefly Algorithm is only applied to function optimization problems in continuous space while TSP is a problem of combinatorial optimization. Therefore, Firefly Algorithm needs to be improved, so that the algorithm can solve the discrete problems of combinatorial optimization. The specific program is divided into the following sections:

3.1 Selection of fireflies flying city

In solving TSPs, after initializing the firefly location, the firefly individual will choose the next city to fly to till covering all cities to form a Hamilton Circuit with the initial city location due to that its initial position cannot constitute a viable solution to the problem. To be easy to describe, this article gives the following definitions:

Definition 5: the set constituted by City *n* and those cities connecting to City *n* is called as *n*'s neighborhood city set, written as *ne*{*n*}.

Definition 6: Firefly *i* has never been to the cities which constitute a set in Loop t, called as the currently selected city set of Firefly *i*, written as *tau*{*t,i*}.

Definition 7: The intersection of *ne*{*n*} and *tau*{*t,i*}, *pc*{*t,i*} is applied when for Firefly *i* is in City *n* and can fly to the next city of the city set.

Completing the initialized position of the firefly individual, the firefly can fly from current City *n* to a certain city in *pc*{*t,i*} to realize location updates. When the Firefly *i* reaches City *n*, judging the fluorescent brightness and path length between City *n* and other cities in *pc*{*t,i*} to choose the next city. In order to enhance the diversity and avoid a local optimum early, this article uses the roulette method to choose the next city. When *tau*{*t,i*} is empty, it means the Firefly *i* has found a Hamilton Circuit in Loop t.

This article has defined the probability formula of cities selected by the firefly:

$$P_{n,j\in pc\{t,i\}} = \frac{ioti(n,j)\cdot\gamma(n,j)}{\sum\limits_{j\in pc\{n,j\}} ioti(n,j)\cdot\gamma(n,j)} \qquad (5)$$

Among which, *n* is the number of current city for the firefly, $P_{n,j}$ is probability of Firefly *i* flying from City *n* to City *j*, *ioti*(*n,j*) is City *n* and City *j* on the path of fluorescent brightness, $\gamma(n,j)$ is the reciprocal value of the path between City *n* and City *j*, that is, $\gamma(n,j)=1/d(n,j)$.

From Eq. (5), we can observe when Firefly *i* selects the next city to fly to, its selection probability is closely associated with the fluorescent intensity and path lengths between cities. Firefly *i* will choose a shorter path, and a city with higher fluorescent intensity.

3.2 Update Mechanism of Fluorescent Intensity

In the Firefly Algorithm, the individual firefly fitness is indicated as the path length of a single trip to all cities for each firefly, that is:

$$fitness=length \qquad (6)$$

Among which, length means the flying path length of the firefly's single trip to all cities.

In Loop t, when all individuals have completed the single trip, calculate firefly individual's path length of flying in a circuit and evaluate the all path solutions, finding out an optimal path solution as Route_best and its path length as length_best, to update the fluorescent light on the path. As update of fluorescent brightness has not been referred in the traditional FA, this paper provides an update mechanism of fluorescent brightness:

$$ioti^{t+1}(m_1,m_2)=(1-\rho)\cdot ioti^{t}(m_1,m_2)+\sqrt{\frac{ioti^{t}(m_1,m_2)\cdot length\_best}{d(m_1,m_2)\cdot\sum ioti^{t}}} \qquad (7)$$

$$ioti^{t+1}(m_2,m_1)=ioti^{t+1}(m_1,m_2) \qquad (8)$$

Among which, the volatile coefficient ρ is the brightness of the fluorescence, volatilization fluorescent brightness, m_1 and m_2, on the path of the cities within the time of [t,t+1] is $(1-\rho)\cdot ioti^{t}(m_1,m_2)$, and $\sqrt{\frac{ioti^{t}(m_1,m_2)\cdot length\_best}{d(m_1,m_2)\cdot\sum ioti^{t}}}$ indicates the increased amount of fluorescent intensity within the time of [t,t+1] . Due to that the fluorescent brightness matrix on the city path is a pair of symmetric matrices, so the $ioti(m_2,m_1)= ioti(m_1,m_2)$.

After initializing the firefly location, firefly individuals choose the location at first until the species complete the single trip to all cities, and update of fluorescent brightness on the city path according to the species' flying path length. In each iterative process, the better location individual firefly has, the shorter the flight path it will take, updating the path of fluorescent brightness can attract more individuals to select the path on the city side in the next iterative flying process, speeding up the convergence rate of the algorithm.

3.3 Firefly step length setting and location update

In the traditional FA, Firefly will complete the Firefly location updates according to the attraction of the higher fluorescent intensity than itself. In solving TSPs, firefly individual need to fly from the current City *n* to the next city. As path length between the cities are different, it is difficult to complete the set of step factor α. Therefore, this paper proposes the following definition:

Definition 8: the distance between the cities is the step length of firefly's movement. As the varying distances between cities, step length factor shall be a dynamic value.

When Firefly *i* reaches City *n*, it will determine the next flight city *j* according to the fluorescent brightness on the city path, changing the value of step length factor, flying to city *j* to complete the location updates and evolution of firefly individuals.

3.4 Mutation Mechanism and Elite Strategies

Traditional Firefly Algorithm will gather individual fireflies together on a later stage and the algorithmic tracking mechanism will lose efficacy and lead the algorithm into a local optimum.

In order to avoid "premature" status, this paper introduces mutation mechanism to determine whether the algorithm gets into a local optimum. If so, exchange any two cities locations; if not, no operation is needed.

Meanwhile, in order to avoid receiving poor solutions after a mutation operation, this paper introduces elite strategies to determine if variation of path length is shorter. If so, accept a mutation path solution; if not, abandon a mutation path solution and keep the original path solution.

Mutation and elite strategies can improve search accuracy. When the algorithm is faced with a local optimum, the algorithm has the ability to get out of the local Optimum.

3.5 Fluorescent intensity catastrophe mechanism

In the later phase of Firefly Algorithm, the big difference between the fluorescent brightness on the edge of the city leads firefly individual to probably choosing the city path with high fluorescent brightness in the city selection process, thus abandoning some better city paths.

In order to weaken the influence of the differences between fluorescent brightness on the path selection of the city and process the fluorescent brightness change on the city path to make re-initialization, enabling firefly individuals to have more choices in city selection with the enhancement of the diversity and acceleration of the convergence rate.

Meanwhile, big differences between fluorescent brightness occurring for many times result in the algorithm to the path solution of local optimum for recording, analysis and comparison and some paths are found to be shared for all solutions. Therefore, these paths are likely to be optimal public paths in the process. The fluorescent brightness is intensified on the public path while keeping others as usual, which will lead a greater probability of selecting the public path to the optimal path direction in the running process. Improved algorithms will construct a optimal path from the city path of non-public paths to make a shortest path with the public paths.

Change mechanism can effectively weaken those phenomena that some better path solutions have to be discarded due to big differences between fluorescent brightness on the edge of cities, enhancing the algorithm diversity.

3.6 Complete 2-Opt (C2Opt) Operators

In Firefly Algorithm to solve TSPs, it is easy to produce intersections, leading to increase path length. Therefore, in each iterative process, the flight paths of firefly individuals needs to have global optimization and removal of the road intersection.

To this end, this paper introduces the C2Opt operator which constitutes a path for covering all cities to have 2-Opt global optimization, Examples are illustrated in Figure 1 and Figure 2 to specify C2Opt operator process.

Among which, the algorithm of Figure 1 has the phenomenon of road intersection; Figure 2 is elimination of the road intersection after using C2Opt operator, elimination of the road intersection.

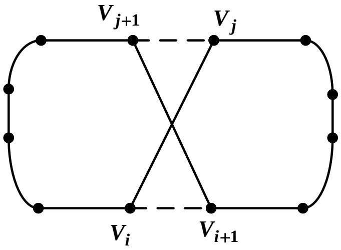

Fig. 1. the Route figure before use C2Opt operator

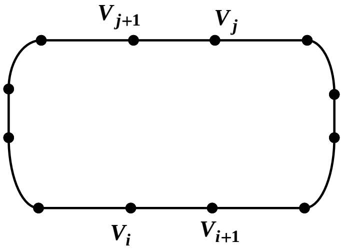

Fig. 2. the Route figure after use C2Opt operator

Suppose, Figure 1 and Figure 2 are the same in addition to combinatorial methods of V_{j+1},V_j,V_i and $V_{i+1.}$Set the connecting line of V_{i+1} and V_{j+1} and crossing point of V_i and V_j as point O in Figure 1 . According to the theorem that the sum of two sides is longer than the third side of the triangle:

$D(V_j,\mathrm{O})+ D(V_{j+1},\mathrm{O})>D(V_j,V_{j+1})$

$D(V_i,\mathrm{O})+ D(V_{i+1},\mathrm{O})>D(V_i,V_{i+1})$

From such a simple reasoning, C2Opt operator can efficiently remove road intersection and shorten the path length. Next, the C2Opt algorithmic process is described as below:

```
BEGIN
FOR $i$ = 1 To $n$ THEN /* n means the total quantity of cities */
    IF $i$ = $n$ OR $n$-1 OR $n$-2 THEN
        $i$=$n$,    exchange city $V_1$ and $V_2$.
        $i$=$n$ -1, exchange city $V_n$ and $V_1$.
        $i$=$n$-2,  exchange city $V_n$ and $V_{n-1}$.
   ELSE
        FOR $j$ = $i$+1 To $n$-1 THEN
            IF  $D_{i,j}+D_{i+1,j+1}<D_{i,i+1}+D_{j,j+1}$ THEN
                Exchang city $V_{i+1}$ and $V_{j.}$
                Interchange points between $V_{i+2}$ and $V_{j-1}$.
END
```

In the running process, when a road intersection occurs to the algorithm, C2Opt operators can effectively eliminate road intersection, accelerating algorithmic optimal rate and improving the accuracy of the search

3.7 FA Process for solving TSP

```
BEGIN
INITIALIZE
The $m_k$ Firefly are randomly placed in the city point with parameters initialization.
Give the same initial fluorescent brightness to each city line.
REPEAT
FOR each firefly in the group THEN
Firefly $i$ will choose the next city flying to according to Eq. (5) by the way of roulette.
Judge if intersection occurs to city paths when Firefly $i$ fly on a single circuit.
If so, using C2Opt operator to remove road intersection.
```

```
   Length_best=min(Length).
   Update optimal path of fluorescent brightness by equation (7) and (8).
   IF  the fluorescent brightness difference on the city path is too large in
the algorithm THEN
   Re-initialize the fluorescent brightness.
   Enhance fluorescence on the public path
   Determine whether the algorithm gets into a local optimum.
   If so, use mutation and accept the shorter path length thereafter.
   UNTIL  the iteration reaches the maximum.
   END
```

4 Analysis of Experimental Results

4.1 Parameter Settings

In computer simulations, TSPLIB instances are used for comparison. The initial fluorescent brightness values of all instances are 10 and other parameters are set as shown in table 1.

Table 1. experimental parameter settings

Exam ple	Att48	Eil51	Berlin 52	St70	Kroa 100	Krob 100	Lin 105	Pr107	Pr124	Pr136	Pr152	Pr226	Lin 318
$ioti_0$	10	10	10	10	10	10	10	10	10	10	10	10	10
ρ	0.3	0.3	0.3	0.32	0.3	0.31	0.31	0.31	0.31	0.31	0.35	0.4	0.5
N	48	51	52	70	100	100	105	107	124	136	152	226	318
M	50	50	50	50	50	50	50	50	50	50	50	50	50
itermax	200	200	200	200	200	200	200	200	200	200	200	200	200

For solving TSPs in different sizes, the parameters in table 1 $ioti_0$, ρ , N , M , $itermax$ are randomly selected with different values through repeated experiments to observe the effect of parameters on optimal results. In all results, the corresponding parameter value of achieved optimal short diameter is put as the experimental parameter value in this paper..

4.2 Results Comparison

In order to verify the reliability of the proposed algorithm in solving TSPs, multiple TSPLIB instances are used to have a test with the test results compared with ones obtained by other algorithms.

Table 2 shows the algorithm instance, Att48, Eil51 and Berlin52 of the experimental results and [11] indicates STA in comparison of the experimental results where the optimal value is the shortest path length provided by TSPLIB.

Table 2. Comparison of this (FA) experimental result and STA result

Example	the best value	FA		STA[11]	
		the best results	error	the best results	error
Att48	33523	33523	-	33724.4	0.6%
Eil51	426	428.8718	0.67%	432.0332	1.42%
Berlin52	7542	7544.36	0.03%	7544.4	0.03%

Note: When the searchable optimal path is less than or equal to the optimal path provided by TSPLIB, the corresponding item is "-"

The calculated formula of deviation rate is:

error=(the best result – the best value)/ best value (9)

As can be seen from table 2 that the algorithm in this paper can search a better path than STA algorithm does in the examples mentioned above.

Though FA fails to search the optimal path or better path provided by the standard library, the result is better and deviation rate is smaller compared to STA. With the increasing city size, the number of iterations required by STA will is proliferating with a larger population size and a larger deviation rate, while FA with small population size in fewer iterations can search better solutions. It can be seen that FA for TSP in small size has better performance than STA.

In order to verify the validity of the algorithm, this paper uses GSO, OMACO and LEO to make performance comparison. Table 3 provides a comparison of searchable optimal result made by the algorithm of this paper and ones made by other algorithms. Optimal values, the best calculated results and deviation rate are indicated the same as those in table 2.

Table 3. experimental result made by the algorithm (FA) compared with one obtained by other algorithms

Example	the best value	FA		GSO		OMACO[12]		LEO	
		the best results	error	the best results	error	the best results	error	the best results	error
St70	675	677.109	0.31%	677.109	0.31%	678.62	0.54%	684	1.33%
Kroa100	21282	21285.4	0.16%	21285.4	0.16%	21320.96	0.18%	N/A	N/A
Krob100	22141	22139.07	-	22139.07	-	22196.53	0.25%	N/A	N/A
Lin105	14379	14383	0.028%	14383	0.028%	14383	0.028%	N/A	N/A
Pr107	44303	44346	0.097%	N/A	N/A	44620.18	0.72%	44521	0.492%
Pr124	59030	59030	-	59030	-	N/A	N/A	N/A	N/A
Pr136	96772	97182.74	0.42%	97684.4	0.943%	N/A	N/A	97212	0.454%
Pr152	73682	73683.6	0.002%	73683.6	0.002%	N/A	N/A	N/A	N/A
Pr226	80369	80369	-	80423	0.067%	80382.45	0.017%	82913	3.17%
Lin318	42029	42440	0.98%	N/A	N/A	N/A	N/A	N/A	N/A

Note the corresponding item "N/A" indicates that the documents does not provide the corresponding value.

It can be seen from Table 3 that the algorithm has found the optimal length of 22139.07 in Krob100 experiment, 1.93 shorter than one as 22141 provided by TSPLIB , Figure 3 is the experimental optimal path of Krob100.

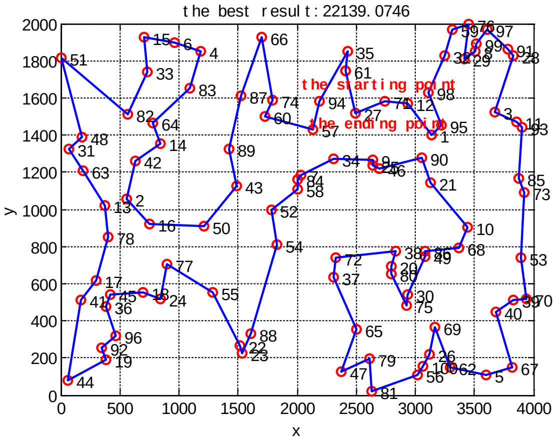

Fig. 3. the experimental optimal path of Krob100

Compared with GSO, LEO and ERS-IWD, in the case of small-size cities, both FA and these algorithms can search for the optimal path. However, with the city size increasing, the algorithms mentioned above is not as good as FA to search the optimal path. Thus, FA is superior to others in both small-size and large-size cities.

Through the compared analysis of experimental results mentioned above, although FA fails to search the shortest path indicated in TSPLIB in some instances, deviation rate is within its tolerance scope as 1% for optimal pathd compared to other algorithms, closing to the standard data mentioned in the library, which means FA have a high precision and speed for solving TSPs.

5. Conclusion

This paper has proposed an improved solution of Firefly Algorithm for solving TSPs. As the traditional algorithm get easily into a local optimum to a problem, the article uses the method of combining mutation mechanism and elite strategies as well as change mechanism of fluorescent brightness to effectively improve the algorithmic accuracy and enhances the diversity. In order to solve the problem of road intersection in the running process, introduction of C2Opt operators in algorithm for global optimization, can greatly shorten the optimal path, accelerating the convergence rate. Through the sample test of TSPLIB, the algorithm is verified as high accuracy and searching speed of optimization with a good applied valued in solving combinatorial optimization and relevant problems.

Acknowledgments

The work was supported by Ningbo Natural Science Foundation (No.2014A610069), Scientific Research Fund of Zhejiang Provincial Education Department (No. Y201326770), Zhejiang Xinmiao Talents Project(No. 2015R405031), the Twelfth Five-Year Plan of Zhejiang province key discipline (Computer application technology),China.

References

[1] Yongquan Zhou, Zhengxin Huang. Artificial glowworm swarm optimization for TSP[J].Control and Decision.2012,27(12):1816-1821.

[2] Basem O.Alijla, Li-Pei Wong, Chee Peng Lim, Ahamad Taijudin Khader,Mohammed Azmi Al-Betar.A modified Intelligent Water Drops algorithm and its application to optimization problems[J].Expert System with Applications.2014,(41)5:6555-6569.

[3] Junzhong Ji, Zhen Huang , Chunnian Liu. An Ant Colony Algorithm Based on Multiple-Grain Representation for the Traveling Salesman Problems[J].Journal of Computer Research and Development.2010,47(3):434-444.

[4] Huafeng Wu, Xinqiang Chen, Qihuang Mao. Improved and colony algorithm based on natural selection strategy for solving TSP problem[J].Journal on Communications. 2013,34(4)： 165-170

[5] Michalis Mavrouniotis, Shengxinang Yang. Ant colony optimization with immigrants schemes for the dynamic traveling salesman problem with traffic factors[J].Applied Soft Computing,2013,13(5):4023-4037.

[6] Thirachit Saenphon, Suphakant Phimoltares , Chidchanok Lursinsap. Combining new Fast Opposite Gradient Search with Ant Colony Optimization for solving travelling salesman problem[J].Engineering Applications of Artificial Intelligence.2014:324-334.

[7] YANG Xin-she，DEB S．Eagle strategy using lévy walk and firefly algorithms for stochastic optimization[J].Studies in Computational Intelligence.2010,28(4): 101-111．

[8] Leandro dos Santos Coelho, Viviana Cocco Mariani. Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning[J]. Computers & Mathematics with Applications, October 2012, 64(8): 2371-2382.

[9] J.Senthilnath,S.N. Omkar, V. Mani. Clustering using firefly algorithm: Performance study[J].Swarm and Evolutionary Computation, 2011, 1(3): 164-171.

[10] Praveen Ranjan Srivatsava, B. Mallikarjun, Xin-She Yang.Optimal test sequence generation using firefly algorithm Original Research Article[J]. Swarm and Evolutionary Computation, 2013,8(1): 44-53.

[11] Chunhua Yang, Xiaolin Tang, Xiaojun Zhou. A discrete state transition algorithm for traveling salesman problem[J].Control Theory & Applications. 2013, 30(8): 1040-1046

[12] Pengzhen Du, Zhenmin Tang, Yan Sun. An object-oriented multi-role ant colony optimization for solving TSP problem[J].Control and Decision. 2014,(29)10:1729-1736