

# Modified Discrete Firefly Algorithm Combining Genetic Algorithm for Traveling Salesman Problem

Lin Teng, Hang Li\*

Software College, Shenyang Normal University, Shenyang, P.C 110034, China

\*Corresponding author, e-mail: lihangsoft@163.com

## Abstract

*The Firefly Algorithm (FA) has a few disadvantages in solving the constrained global optimization problem, including that it is difficult to produce initial population, the size of relative attractiveness has nothing to do with the absolute brightness of fireflies, the inertia weight does not take full advantage of the information of objective function, and it cannot better control and constrain the mobile distance of firefly. In this paper, we propose a novel method based on discrete firefly algorithm combining genetic algorithm for traveling salesman problem. We redefine the distance of firefly algorithm by introducing swap operator and swap sequence to avoid algorithm easily falling into local solution and accelerate convergence speed. In addition, we adopt dynamic mechanism based on neighborhood search algorithm. Finally, the comparison experiment results show that the novel algorithm can search perfect solution within a short time, and greatly improve the effectiveness of solving the traveling salesman problem, it also significantly improves computing speed and reduces iteration number.*

**Keywords:** firefly algorithm, traveling salesman problem, genetic algorithm, neighborhood search algorithm

Copyright © 2018 Universitas Ahmad Dahlan. All rights reserved.

## 1. Introduction

Traveling salesman problem (it is abbreviated in TSP) [1] is an important combinatorial optimization problem in the area of mathematics. It belongs to Non-Deterministic Polynomial (NP) problem. Although there are some precise algorithms which can be used to solve the problem, the principle of precise algorithms is complex, and it can produce “combination explosion problem” along with the increase number of city, therefore, the domestic and foreign scholars have been trying to seek a highly efficient and stable algorithm for solving this complex problem. For small size TSP [2], Branch and bound method, greedy method and cutting plane method are often used to solve these problems. But for big size TSP, it is difficult to use above methods. Therefore, the domestic and foreign scholars have been trying to seek a highly efficient and stable algorithm for solving this complex problem. Firefly algorithm (FA) [3] is one of heuristic search algorithms based on swarm intelligence, which was proposed by YANG [4] in 2010. FA [5] is based on the self-organization of the swarm simulation model with the advantages of less setting parameters, strong robustness, and has been applied in many fields. Li et al. [6] showed that firefly algorithm was introduced to solve this problem, and a series of discrete operations were conducted to adapt to it. Yin et al. [7] proposed discrete artificial bee colony algorithm for TSP. Zhang [8] presented a novel firefly algorithm for solving the nonlinear equations problem.

The above methods almost are used for solving continuous domain optimization problems. However, FA is relatively few used in the aspect of discrete domain application. To improve the performance of TSP solution, we propose improved discrete FA through redefining distance of firefly algorithm by introducing swap operator and swap sequence which better coordinates and balances the search process of FA algorithm. To facilitate the description, this paper also gives some definitions. We present a new discrete FA with genetic algorithm (GA) for traveling salesman problem. This new scheme takes neighborhood search algorithm into consideration. Finally, we conduct some experiments to verify its performance. The results show that the new algorithm has a good effect on solving TSP.

## 2. The Firefly Algorithm

Fireflies emit fluorescence signal in the nature. From the point of biology view, its role is to attract the opposite sex and get copulation, reproduction chance, sometimes for predation. Firefly uses fluorescence signal within a certain range to attract companion in FA, eventually it makes most fireflies gather in one or more locations, and realize the position optimization.

Fireflies attract companies mainly depending on fluorescence intensity and attraction degree. Fluorescence intensity depends on the target value of its own location. The better target value is, the high fluorescence intensity is. Attraction degree has closely relationship with fluorescence intensity. Brighter fireflies have higher attraction intensity, which can attract weaker fireflies moving towards to it. In addition, it should take physical properties into consideration in FA. Fluorescence intensity and attraction degree are in inverse proportion of the distance among fireflies. Namely, fluorescence intensity and attraction degree will decrease with the increase of distance.

Fireflies mainly obey the following rules:

- a. firefly moving direction is determined by the fluorescence intensity, it always move towards to the brighter fireflies.
- b. the firefly moving distance is determined by the attraction degree.

This paper defines firefly fluorescence intensity and attraction degree as follows.

Definition 1.  $I$  : Fluorescent brightness of fireflies.

$$I = I_0 e^{-\gamma r_{ij}} \quad (1)$$

Where  $r_{ij}$  is the distance between firefly  $i$  and  $j$ .  $\gamma$  is light intensity absorption coefficient, it denotes that fluorescence gradually decreases with the increase of distance, it can be set as a constant.  $I_0$  is fluorescent brightness of firefly when  $r = 0$ .  $I_0$  not only is its own fluorescence intensity, but is the maximum fluorescence intensity, which has relationship with function value.

Definition 2.  $\beta_{ij}$  : Attraction degree of fireflies.

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2} \quad (2)$$

Where  $\beta_0$  (maximum attraction degree) is attraction degree of firefly when  $r = 0$ . According to definition 1, every firefly uses roulette method to move towards to the individual with higher fluorescence intensity. On the basis of definition 2, each firefly can determine the movement distance. By constantly updating its location, it achieves the purpose of optimization.

Definition 3. Location updating formula of firefly  $i$  attracted by firefly  $j$ .

$$x_i = x_i + \beta(x_j - x_i) + \alpha(rand - 0.5) \quad (3)$$

This formula is composed of three parts. First part,  $x_i$  denotes current position of firefly  $i$ . Second part,  $\beta(x_j - x_i)$  denotes the distance firefly  $i$  moving to  $j$ . Third part, there is disturbance mechanism when location updating to avoid fireflies prematurely falling into local optimal solution.  $\alpha \in [0,1]$  is step length factor.  $rand \in [0,1]$  is random factor and obeys uniform distribution.

## 3. The Modified Firefly Algorithm for TSP

We first use genetic algorithm to initialize population generation. Assuming that initial group size is  $m$ , and it randomly generates  $m$  individuals. The  $i$ -th is  $X_i^t$ . For each individual,

it determines the index set  $S_t$  and  $T_t$ .  $T_t = \{j | g_i(X'_i) > 0, 1 \leq j \leq l\}$  and  $S_t = \{j | g_i(X'_i) \leq 0, 1 \leq j \leq l\}$ , where  $T_t$  and  $S_t$  satisfy the constraint condition.

When we solve TSP by FA, we should know distance and light intensity absorption coefficient. Then it updates location through formula (1,2,3) to realize optimizing. We apply serial number coding method into FA to solve TSP in this paper. Because TSP is a combinatorial optimization problem, we should first definite discrete variable distance, then we present discrete variable location updating formula and its disturbance mechanism. Finally, we give the solution step based on discrete firefly algorithm for solving TSP.

In this section, we introduce the following new definitions.

Definition 4. Assuming that solution sequence of  $n$  nodes TSP is  $S = (a_i)$ , swap operator  $S_0(i_1, i_2)$  is the point  $a_{i_1}$  and  $a_{i_2}$  in swap solution  $S$ . So  $S' = S + S_0(i_1, i_2)$  is the new solution after operation with swap operator.

Example 1. There is a five nodes TSP, its solution is  $S = (2 \ 1 \ 3 \ 5 \ 4)$ . Swap operator is  $S_0(2,3)$ . So  $S' = S + S_0(i_1, i_2) = (2 \ 1 \ 3 \ 5 \ 4) + S_0(2,3) = (2 \ 3 \ 1 \ 5 \ 4)$ .

Definition 5. Ordered queue of one or multiple swap operators is called a swap sequence.

$S' = (S_{o_1}, S_{o_2}, \dots, S_{o_n})$ ,  $S_{o_1}, S_{o_2}, \dots, S_{o_n}$  is swap operator.

Definition 6. Different swap sequences act on the same solution that may generate the common new solution. Therefore, all the same effect swap sequences set is called equivalent set of swap sequence.

Definition 7. In the equivalent set of swap sequence, the swap sequence with minimum swap operators is the basic swap sequence in equivalent set.

Space distance between fireflies can be calculated by Euclidean distance when solving optimization problems of continuous variables. However, when solving TSP, solution is discrete variable, so it cannot use Euclidean distance. In this paper, we definite space distance:

$$r = A / N \quad (4)$$

Where  $A$  is swap operator number of two fireflies in solution vector basic swap sequence.  $N$  is the cities number. After getting  $r$ , fluorescent brightness of firefly can be calculated by formula (1). Therefore, each firefly can move towards to other individuals with higher fluorescent brightness based on roulette wheel method.

Firefly movement distance can be calculated according to attraction degree when solving optimization problems of continuous variables. However, when solving TSP, movement is discrete variable, so we definite the distance between firefly  $i$  and  $j$ :

$$\beta_{ij} = \text{rand int}(0, A_{ij}) \quad (5)$$

Where  $\text{rand int}(0, A_{ij})$  is random integer.  $A_{ij}$  is swap operator number of two fireflies  $i$  and  $j$  in solution vector basic swap sequence.  $\beta_{ij}$  is random swap operator number from 0 to  $A_{ij}$ . Location updating by formula (6):

$$x_i = x_i + \beta_{ij} \quad (6)$$

When solving the TSP, in order to avoid the firefly algorithm falling into local optimum, this paper defines the disturbance mechanism of discrete variables based on variable neighborhood search algorithm. It adopts several neighborhood structure forms to search optimization solution, systematically changes its neighborhood so as to expand the search scope, enhance the ability to jump out of local optimal.

In this paper, we adopt three neighborhood structures to solve TSP.

- a. *Insert* neighborhood. We randomly select different position  $x$  and  $y$  in solution sequence, and insert the city in  $x$  into  $y$ , as  $Insert(x, y)$ .
- b. *Swap* neighborhood. We randomly select different position  $x$  and  $y$  in solution sequence, and exchange the cities in  $x$  and  $y$ , as  $Swap(x, y)$ .
- c. *2-opt* neighborhood. We use side  $(i, j)$  and  $(i+1, j+1)$  to replace  $(i, i+1)$  and  $(j, j+1)$ . This change will affect the direction of mid-edge  $(i+1, j)$ .

The detailed process of the proposed discrete firefly algorithm is as follows:

Step 1. Initializing the swarm. Set parameter firefly number  $m$ , light intensity absorption coefficient  $\gamma$ , maximum iteration number  $T_{max}$ .

Step 2. Randomly initializing solution sequence of each firefly. Calculating objective function  $P_i$  and maximum fluorescent brightness  $I_0$ .

$$I_0 = P_g / P_i \quad (7)$$

Where  $P_g$  is the current searched optimal solution.

Step 3. According to formula (1), (4), (5), (7), calculating relative brightness  $I$  of firefly and attraction degree  $\beta_{ij}$ . So determine the movement direction and distance.

Step 4. According to formula (6), updating position of firefly.

Step 5. Choosing the three neighborhood in variable neighborhood search method based on a certain probability. Repeat executing  $n$  iteration. Selecting position with optimal objective function as the current position of firefly.

Step 6. According to updated position of firefly, re-compute objective function  $P_i$  and make a comparison with current optimal position  $P_g$ . If function is better, then update  $P_g$ .

Step 7. Whether the process satisfies termination conditions or not. If YES, then finish and output results. If NO, return back step3.

## 4. Experiments and Results

### 4.1. The effect of Light Intensity Absorption Coefficient $\gamma$ on New Firefly Algorithm

The light intensity absorption coefficient  $\gamma$  is the main parameter in new firefly algorithm with GA (abbreviate in FGA). We make simulation experiments to analyze the effect of  $\gamma$  on performances of algorithm solution under the MATLAB platform. We adopt different  $\gamma$  to conduct experiments under the other same parameters conditions. Setting maximum iterations  $T_{max} = 500$ , number of firefly is  $m = 50$ , the three neighborhood in variable neighborhood search method are same:  $1/3$ , algorithm runs 30 times independently. Table1 records the results with different  $\gamma$ .

Table 1. Different  $\gamma$  with Different Influences on FGA

$r$	Best value $\times 10^3$	Worst value $\times 10^3$	Average value $\times 10^3$	Standard deviation $\times 10^2$
0.01	7.51443	8.48821	8.01094	2.34474
0.03	7.52443	8.48449	8.00774	2.07446
0.05	7.52466	8.58599	8.10785	2.48271
0.07	7.63768	8.68004	8.07167	2.38243
0.09	7.53436	8.59042	8.10332	2.17915
0.11	7.52436	8.69427	8.22529	3.19748
0.13	7.73988	8.78536	8.17088	2.49888
0.15	7.75758	8.86203	8.35733	2.68301

From Table 1 we can know that when  $\gamma$  gradually increases, the solution performance is little-changed. When  $\gamma=0.01, 0.02, 0.03, 0.04, 0.06, 0.09$  and  $0.11$ , FGA gets the best solution,  $7.53436 \times 10^3$ . And  $\gamma=0.03$ , average value is  $8.00774 \times 10^3$  and standard value  $2.07446 \times 10^2$  which is the minimum value. But as a whole, when  $\gamma=0.03$ , FA has a better solving performance, which is the best value for FGA. The effect of different  $\gamma$  on FA is not obvious. So this is one of advantages for FGA.

#### 4.2. The Comparison Experiment

In order to further verify the feasibility and effectiveness of this paper's method, we choose multiple instances in international universal test library for testing, and use testing results to make a comparison to spanning tree algorithm (STA) [9], Improved Ant Colony Optimization [10](IACO).

In the experiment, digit indicates number of cities. "O" shows calculation result of the index is not given in the comparison literature. Experimental environment is the same as above. Parameters set as follows: maximum number of iterations  $T_{\max} = 2000$ ,  $\gamma = 0.03$ ,  $k_1:k_2:k_3 = 2:1:2$ , number of firefly  $m = 20$  when city number is less than 48, otherwise  $m = 50$ . The two algorithms run 20 times. Calculation results are as shown in Table 2.

Table 2. Calculation Results

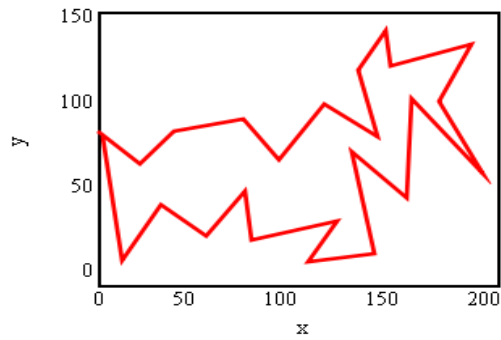
Instance	Calculation index	IABC	STA	FA
Bays29	Best value	30.8788	30.8788	30.8788
	Average value	30.8788	30.8788	30.8788
	Iteration number	2000	200	200
	Running time	5.11	O	O
Oliver30	Best value	73.987	73.9875	73.9875
	Average value	74.0079	74.0077	74.0075
	Iteration number	2000	200	200
	Running time	5.36	O	O
Dantzig42	Best value	76.1236	76.1149	75.9098
	Average value	75.3069	75.3069	75.3069
	Iteration number	2000	200	200
	Running time	8.95	O	O
Eil51	Best value	431.25	428.88	426.01
	Average value	444.19	431.01	427.50
	Iteration number	2000	200	200
	Running time	10.36	O	O
Berlin52	Best value	7680.77	7554.38	7542.13
	Average value	7941.26	7562.81	7573.49
	Iteration number	2000	200	200
	Running time	10.76	O	O

From Table 2, STA algorithm gets the known optimal solutions in Berlin52, Oliver30 and Eil51, which is close to FGA. But iteration number of FGA has reduced by 98percent. What's more, steps of FGA are not complicated. The average value has increased by 0.6% (Not exceed 0.6%). Compared with IABC algorithm, the best value and average value is relatively small, and for Bays29, Oliver30 and Dantzig42, the standard deviation is smaller. For the four instances, the running time of FGA has saved by 97.1%, 98.2%, 98.8% and 98.99% respectively.

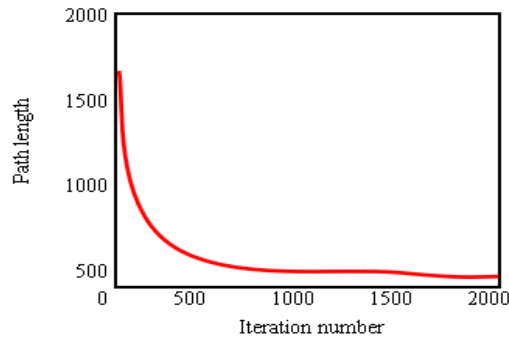
FGA algorithm also can obtain the results closely to known optimal solutions for all instances. For Oliver30 and Bays29, FGA algorithm gets the same results as known optimal solutions. For Dantzig42, calculation result of FA algorithm has decreased by 19.3% compared with IABC algorithm. Figure 1-3 show the experimental simulation diagram of Oliver30, Dantzig42 and Berlin52.

The average value of FGA algorithm is relatively small compared with IABC algorithm for Oliver30, Dantzig42 and Bays29. IABC algorithm is mainly based on the STA algorithm and has combined with other operations. When the number of population in IABC is equal to that in STA, the running time in IABC is closely to that of STA at least. Nevertheless, the running time

in IACO is relatively long. Therefore, FGA algorithm has the better solution performance in terms of solution time and solution quality compared with STA and IABC algorithm.

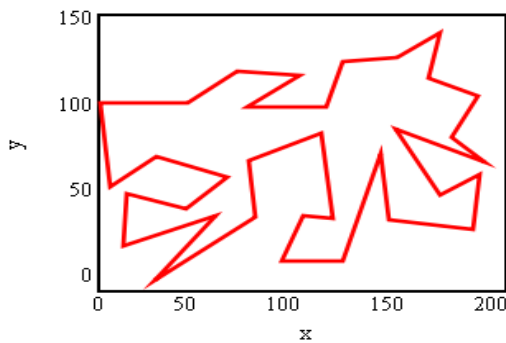


(a) Optimal path

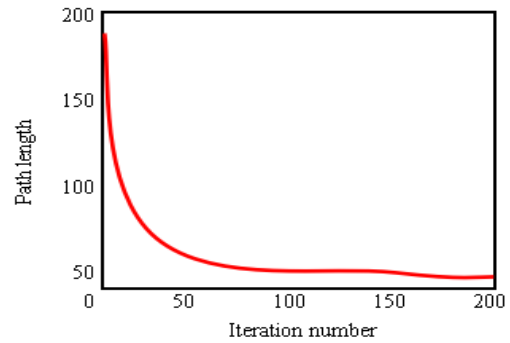


(b) Best value evolution curve

Figure 1. Experimental simulation diagram of Oliver30

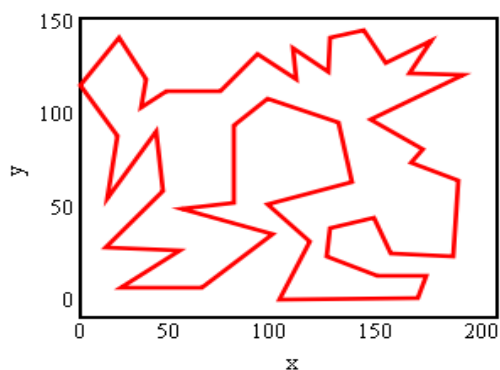


(a) Optimal path

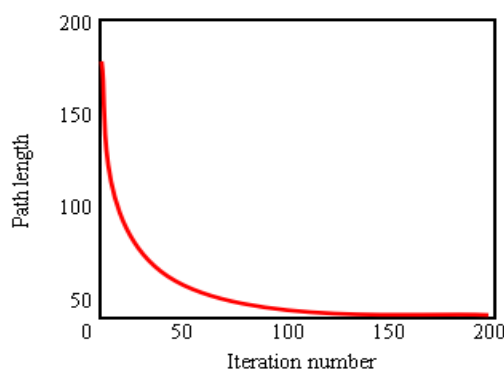


(b) Best value evolution curve

Figure 2. Experimental simulation diagram of Dantzig42



(a) Optimal path



(b) Best value evolution curve

Figure 3. Experimental simulation diagram of Berlin52

## 5. Conclusion

This paper proposes a new discrete firefly algorithm for solving TSP problem. It extends continuous artificial swarm (especially FA) optimization algorithm to discrete domain. The new algorithm makes some definitions based on swap operator and swap sequence. It also adopts disturbance strategy based on variable neighborhood search method to increase search areas and improve the local refinement ability of the algorithm and the optimal speed. The experimental results show that the algorithm can find relatively satisfactory solution in a short time and improve the efficiency of solving the TSP. In the future, we would find more effective firefly algorithms to improve optimum searching method and solve other combinatorial optimization problems.

## References

- [1] Carrabs F, Cordeau J F, Laporte G. Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading [J]. *Inform Journal on Computing*. 2015; 19(4): 618-632 2007.
- [2] Purbasari A, Suwardi I S, Santoso O S, et al. Data Partition and Communication On Parallel Heuristik Model Based on Clonal Selection Algorithm [J]. *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*. 2015; 13(1): 193-201.
- [3] Yang X S. Cuckoo Search and Firefly Algorithm: Theory and Applications [J]. *Engineering Optimization*. 2014.
- [4] Yang, Xin-She. Firefly Algorithm [M]// *Engineering Optimization*. 2010; 221-230.
- [5] Massan S U R, Wagan A I, Shaikh M M, et al. Wind turbine micrositeing by using the firefly algorithm [J]. *Applied Soft Computing*. 2014; 27(2):450-456.
- [6] Li M, Ma J, Zhang Y, et al. Firefly Algorithm Solving Multiple Traveling Salesman Problem [J]. *Journal of Computational & Theoretical Nanoscience*. 2015; 12(7).

- [7] Lei M, Shoulin Y, Xinyuan H. A New Method Used for Traveling salesman problem Based on Discrete Artificial Bee Colony Algorithm. *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*. 2016; 14(1): 342-348.
- [8] Zhang Y F, Li Y G. An Improve Firefly Algorithm and its Application in Nonlinear Equation Groups [J]. *Advanced Materials Research*. 2014; 1049(5):1670-1674.
- [9] Iwata S, Newman A, Ravi R. Graph-TSP from Steiner Cycles [M]// *Graph-Theoretic Concepts in Computer Science*. Springer International Publishing. 2014; 312-323.
- [10] Yu L, Li M, Yang Y, et al. An Improved Ant Colony Optimization for Vehicle Routing Problem [C]// *Logistics@sThe Emerging Frontiers of Transportation and Development in China*. ASCE, 2015:3360-3366.