Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem

Gilang Kusuma Jati and Suyanto

The faculty of Informatics - Telkom Institute of Technology, Jl. Telekomunikasi No. 1 Terusan Buah Batu, Bandung 40257, West Java, Indonesia
gilang.kusuma@live.com, suy@ittelkom.ac.id

Abstract. This paper addresses how to apply firefly algorithm (FA) for travelling salesman problem (TSP). Two schemes are studied, i.e. discrete distance between two fireflies and the movement scheme. Computer simulation shows that the simple form of FA without combination with other methods performs very well to solve some TSP instances, but it can be trapped into local optimum solutions for some other instances.

Keywords: evolutionary firefly algorithm, travelling salesman problem, discrete distance, movement scheme.

1 Introduction

Traveling Salesman Problem (TSP) is one of the most intensively studied problems all round the world. TSP is although looking very simple problem but it is an important problem of the classical optimization problems that are difficult to solve conventionally. Basically in this problem a salesman needs to visit each city one time and returns back to the city from the start point of travelling. Exact completion on this issue will involve algorithms that require seeking the possibility of all the existing solutions so this problem is also belonging to the class of "NP-Complete" problems. As a result, execution time complexity of this algorithm will be exponential to the size of the input given.

It is very hard to accurately solve TSP. Many methods, which belong to evolutionary computations, have been proposed to solve TSP. Some of them are: Memetic algorithm proposed by Luciana Buriol [1], discrete fuzzy PSO by N. Salmani Niasar [3], genetic algorithm combined with ant colony system by Marcin L. Pilat [4], improved bee colony optimization with frequency-based pruning by Li-Pei Wong [11], and an advanced method called heterogeneous selection evolutionary algorithm (HeSEA) proposed by Huai-Kuang Tsai [10]. Those researchers proposed methods that are combinations of a metaheuristic algorithm with a local search or other metaheuristic algorithms. In [1], [3], [4], and [11], the researchers focused on small TSP with hundreds of cities (nodes). Generally, accuracies of the methods are very high where the produced solutions are very close to the known optimum solutions. In [10], the researchers focused on large TSP with thousands of cities. The method they proposed, called HeSEA, is capable of solving a TSP with up to 3,038

A. Bouchachia (Ed.): ICAIS 2011, LNAI 6943, pp. 393–403, 2011.

cities with deviation of 0% compare to the known optimum solution. It also solved a TSP of 13,509 cities with deviation only 0.74 %.

Firefly algorithm (FA) is one of the nature-inspired metaheuristic algorithms developed by Xin-She Yang [7], originally designed to solve continues optimization problem [2], [8]. However, FA can be discretized to solve a permutation problem, such as flow shop scheduling problems [5]. In this research, evolutionary discrete FA (EDFA) is proposed to solve TSP. Two schemes are studied, i.e. discrete distance between two fireflies and the movement scheme. This study is focused on the simple form of FA without combination with any other method. Some TSP instances studied here are the small ones with up to 666 cities. However, some ideas to improve the FA also slightly discussed at the end of this paper, but no deep analysis.

2 Evolutionary Discrete Firefly Algorithm

Nature-inspired methodologies are among the most powerful algorithms for optimization problems. FA is a novel nature-inspired algorithm inspired by social behavior of fireflies. Firefly is one of the most special, captivating and fascinating creature in the nature. There are about two thousand firefly species, and most fireflies produce short and rhythmic flashes. The rate and the rhythmic flash, and the amount of time form part of the signal system which brings both sexes together. Therefore, the main part of a firefly's flash is to act as a signal system to attract other fireflies. By idealizing some of the flashing characteristics of fireflies, firefly-inspired algorithm was presented by Xin-She Yang [7]. Firefly-inspired algorithm uses the following three idealized rules: 1) all fireflies are unisex which means that they are attracted to other fireflies regardless of their sex; 2) the degree of the attractiveness of a firefly is proportion to its brightness, thus for any two flashing fireflies, the less brighter one will move towards the brighter one and the more brightness means the less distance between two fireflies. If there is no brighter one than a particular firefly, it will move randomly; and 3) the brightness of a firefly is determined by the value of the objective function [7]. For a maximization problem, the brightness can be proportional to the value of the objective function. Other forms of brightness can be defined in a similar way to the fitness function in genetic algorithms [8].

Based on [8], FA is very efficient in finding the global optima with high success rates. Simulation by Xin-She Yang shows that FA is superior to both PSO and GA in terms of both efficiency and success rate [8]. Lukasik and Zak also study FA for continuous constrained optimization task. Their experiment demonstrates the efficiency of FA [2]. These facts give inspiration to investigate how optimum FA in solving TSP. The challenges are how to design discrete distance between two fireflies and how they move for coordination.

2.1 The Representation of Firefly

A solution representation for the TSP is a permutation representation as illustrated by Figure 1. Here, a firefly represents one solution. It is just like a chromosome that represents an individual in genetic algorithm. In this representation, an element of array represents a city (node) and the index represents the order of a tour.

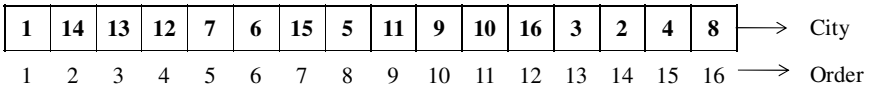

Fig. 1. The permutation representation of a solution

2.2 Distance

In continuous optimization problem, distance between two fireflies is simply calculated using Euclidian distance. For TSP, distance between any two fireflies *i* and firefly *j* can be defined as the number of different arc between them. In Figure 2, three arcs 12-7, 6-15, and 5-11 in firefly *i* do not exist in firefly *j*. Hence, the number of different arcs between firefly *i* and firefly *j* is three. Then, the distance between two fireflies is calculated using formula

$$r = \frac{A}{N} \text{ x } 10, \tag{1}$$

where *r* is the distance between any two fireflies, *A* is the total number of different arcs between two fireflies, and *N* is number of cities. The formula scales *r* in the interval [0, 10] as *r* will be used in attractiveness calculation.

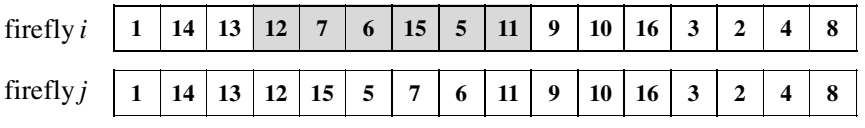

Fig. 2. The distance between two fireflies *i* and *j* is defined as the number of different arcs between them

2.3 Attractiveness

In the original FA, the main form of attractiveness function $\beta(r)$ can be any monotonic decreasing function

$$\beta(r) = \beta_0 e^{-\gamma r^2}, \tag{2}$$

where r is the distance between two fireflies, β_0 is the attractiveness at $r = 0$, and γ is a fixed light absorption coefficient. This scheme is completely adopted by EDFA.

2.4 Light Absorption

In essence, light absorption coefficient γ characterizes the variation of attractiveness value of firefly. Its value is very important in determining the speed of convergence and how the FA behaves. In theory, $\gamma \in [0, \infty)$, but in practice γ is determined by the characteristics of the problem to be optimized.

In condition where $\gamma \to 0$, the attractiveness will be constant and $\beta = \beta_0$. In this case, the attractiveness of a firefly will not decrease when viewed by another. If $\gamma \to$

∞, this means the value of attractiveness of a firefly is close to zero when viewed by another firefly. It is equivalent to cases where the fireflies fly in a very foggy region randomly. No other fireflies can be seen, and each firefly roams in a completely random way. Therefore, this corresponds to the completely random search method.

The coefficient γ functions to determine how much light intensity changes to the attractiveness of a firefly. In this research, γ is in the interval [0.01, 0.15] so that the attractiveness of a firefly viewed by the others will follow the Figure 3.

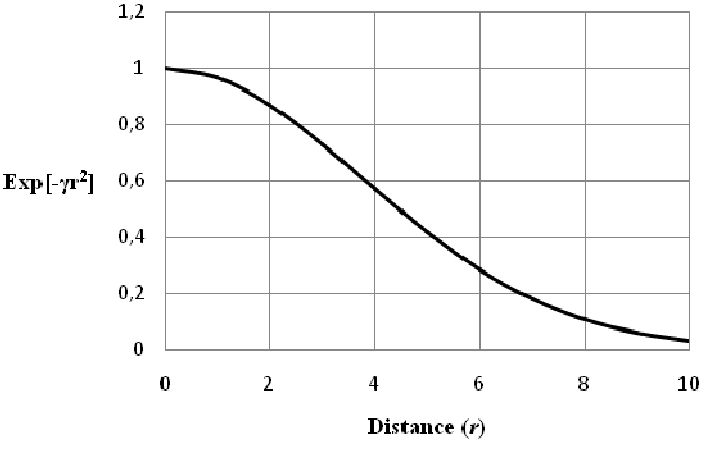

Fig. 3. The correlation of distance and attractiveness

2.5 Movement

The movement of a firefly *i* attracted to another brighter (more attractive) firefly *j* is determined by

$$x_i = random(2, r_{ij}), \tag{3}$$

where r_{ij} is distance between firefly *i* and *j*. The length of movement of a firefly will be randomly selected from 2 to r_{ij}. When a firefly moves, existing solutions in the firefly is changed. Since the representation of firefly is a permutation representation, then we use Inversion Mutation to represent the movement. With inversion mutation, the path that has been formed can be maintained so the good path formed previously is not damaged.

Actually, firefly in EDFA has no direction to move. Hence, it moves using Evolution Strategies (ES) concept. Each firefly will move using inversion mutation for *m* times. First, index on the chromosome will be selected randomly, after it carried an inversion mutation. In other words, each firefly will have *m* new solutions. After *n* fireflies move and produce *n* x *m* new solutions, then *n* best fireflies will be selected as the new population.

2.6 EDFA Scheme

The scheme of EDFA is illustrated by the following pseudo code. First, each firefly generates an initial solution randomly. For each firefly, find the brightest or the most attractive firefly. If there is a brighter firefly, then the less bright firefly will move towards the brighter one and if there is no brighter one than a particular firefly, it will move randomly. When a firefly moves, existing solution produced by the firefly is changed. Each firefly move as much as m times. So, there will be (m x n) + 1 fireflies at the end of iteration since only the best firefly will be included in selection process for the next iteration. Then, n best fireflies will be chosen based on an objective function for the next iteration. This condition will continue until the maximum iteration is reached.

```
Input:
Objective function f(x), x = (x1,... xd)^T                    {cost function}
Initialize a population of fireflies xi (i = 1, 2, ..., n)
Define light absorption coefficient γ and number of moves m   {parameters}
Output:
xi min
begin
    for i = 1 to n do
        xi  Generate_Initial_Solution
    endfor
    repeat
        for i = 1 to n do
            xj  Find_Attractive_Firefly(xi)
            if (xj ≠ null) then
                Move_Firefly(xi, xj) for m times      {move firefly i towards j}
            else
                Move_Random(xi) for m times           {firefly i move randomly}
            endif
        endfor
        Select n brightest fireflies from (m x n) + 1
    until stop condition true
end
```

3 Results and Discussions

In this study, EDFA is applied for 7 TSP instances downloaded from TSPLIB [6]. Table 1 lists the problem names, numbers of cities, and the lengths of the optimal tour. In [6], the types of TSP instances are Euclidian distances. A TSP instance provides some cities with their coordinates. The number in the name of an instance represents the number of provided cities. For example, ulysses16 provides 16 cities with their coordinates. The problem is what the best tour to visit the 16 cities, according to their Euclidian distances, with a condition where each city should be visited only once.

Table 1. Summary of 7 TSPs taken from TSPLIB: problem names, number of cities (nodes) and the length of the optimal tour

Problem names	Number of cities	Length of the optimal tour
ulysses16	16	6859
ulysses22	22	7013
gr202	202	40160
tsp225	225	3845
a280	280	2578
pcb442	442	50778
gr666	666	294358

3.1 Firefly Population

Population size (n) critically determine the computation time. Here, EDFA is tested using various population sizes on problem gr202 to investigate its correlation with number of trials needed by EDFA to get the optimum solution. Figure 4 represents the correlation of the population size with the average trials to reach the optimum solution (with accuracy of 100%). Average trial decreases when the population size is increased from 5, 10 and 15. But, the average trial increases when the population size is 20 or more. Large population does not guarantee that firefly will reach best solution more quickly. According to Figure 4, the best population size is 15.

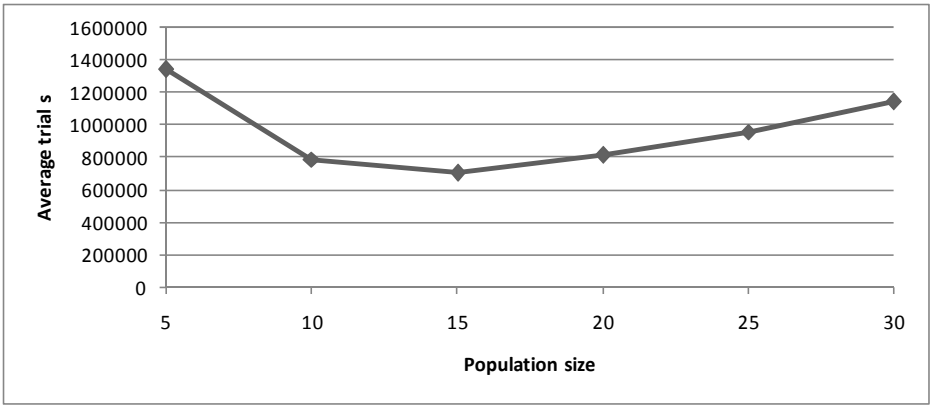

Fig. 4. Correlation of the population size and the average trials needed by EDFA to get the optimum solution

The number of population size determines the number of solutions in each generation. In finding the solution, a firefly with less bright light will follow another one with brighter light (better solution). But, in one generation there could be some fireflies having the same solutions so that the movement of a firefly does not generate better solution.

3.2 Effects of Light Absorption

Light absorption (γ) does not critically determine the computation time. Various values of light absorption were tested on problem gr202 to evaluate its correlation with number of trials needed by EDFA to get the optimum solution. Figure 5 illustrates the relationship between the population size and the average trials to get the optimum solution. Any light absorption, from 0.01 to 0.15, gives quite similar average trials. Thus, the light absorption does not significantly affect the average trials of EDFA.

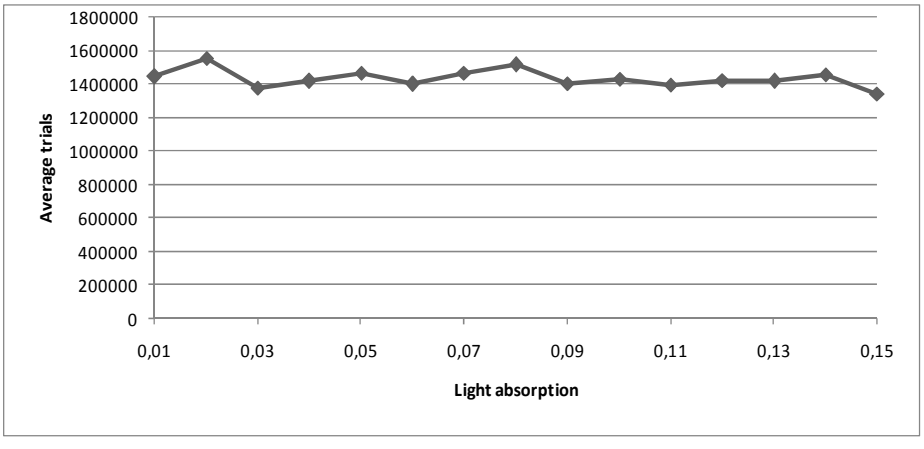

Fig. 5. Correlation of the light absorption and the average trials needed by EDFA to get the optimum solution

3.3 Number of Moves

In EDFA, fireflies have no direction to move. Hence, fireflies move based on a way like in evolution strategies concept. As default, each firefly will make the process of inversion mutation for 8 times. Based on the previous result, the best average trials for problem gr202 can be reached when the population size is 15 and the number of moves is 8. Thus, the total number of moves for each generation is 120. Various numbers of moves, illustrated by Table 2, are tested on problem gr202 to evaluate its correlation with number of trials needed by EDFA to get the optimum solution. The simulation is run for 30 times for each number of moves to get the average trials.

The results show that, in the beginning, theïaverage trials decrease as the number of moves increase. But, when the number of move is 12 or above, the average trials increase. According to Table 2 and Figure 6, the best number of moves is 11 with population size is 11. This setting parameter means that on each generation there will be 11 fireflies and each fireflies move 11 times. The other number of moves, in the interval of 7 and 10, are considerable as the results are quite similar.

Table 2. Results of EDFA applied to gr202 with total trials around 120 in one generation

Problem name	Number of moves	Population size	Total moves per generation	Average trials
gr202	4	30	120	1,342,200
	5	24	120	1,070,808
	6	20	120	854,568
	7	17	119	717,974
	8	15	120	704,232
	9	13	117	656,405
	10	12	120	630,096
	11	**11**	**121**	**586,898**
	12	10	120	694,920
	13	9	117	763,261
	14	9	126	681,609
	15	8	120	662,808

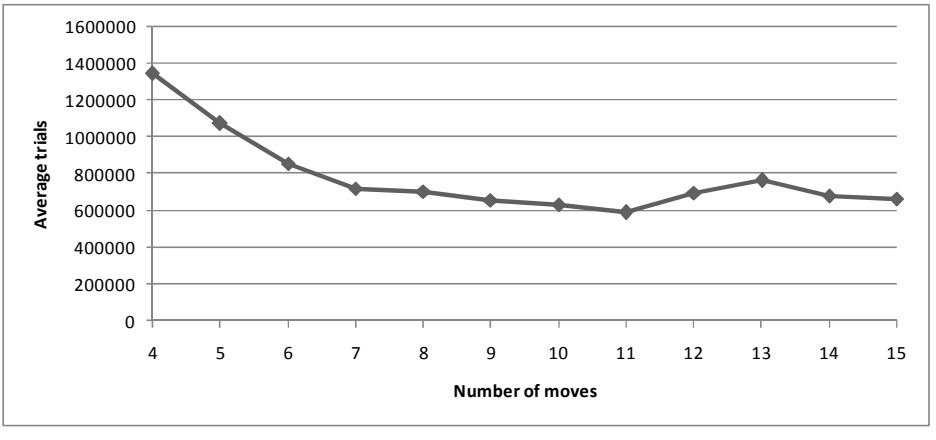

Fig. 6. The correlation of the number of moves with the average trials needed by EDFA to reach the known optimum solution

3.4 Performance of EDFA

EDFA is examined to solve 7 TSP instances to see its performance. In this research, EDFA uses population size of 15 fireflies and the number of light absorption is 0.03. Using those parameters, EDFA are examined using 7 TSP instances. Table 3 shows the worst, best and average accuracy for the 7 instances. The accuracy is calculated by a formula in Equation (4). EDFA always get the best solution for ulysses16, ulysses22, gr202 and gr666 for every runs. However, EDFA is not optimal for three instances: tsp225, a280 and pcb442.

$$r = \frac{best\ solution\ known}{best\ solution\ found}\ x\ 100\% \tag{4}$$

Table 3. Results of EDFA applied to 7 TSP instances, 30 runs for each instance

Problem names	Best solution known	Accuracy (%) Worst	Best	Average	Average time (second)
ulysses16	6859	100.000	100.204	**100.119**	0.416
ulysses22	7013	100.000	100.211	**100.207**	6.590
gr202	40160	100.000	100.653	**100.474**	51.167
tsp225	3845	87.758	89.065	88.332	412.274
a280	2578	87.995	89.668	88.297	691.886
pcb442	50778	87.556	89.457	88.505	3404.211
gr666	294358	100	100.356	**100.033**	393.025

3.5 Comparison of EDFA with Memetic Algorithm

Various studies show that Memetic algorithm can find 76% optimal solution for travelling salesman problem [1]. Now the EDFA will be compared to Memetic algorithm for 7 TSP instances. EDFA, implemented using visual C# .Net, is run 30 times. The results are summarized in Table 4.

Table 4. The comparison between EDFA with Memetic algorithm

Problem names	EDFA Opt/Runs	Memetic algorithm (k=5%) Opt/ Runs
ulysses16	30/30	30/30
ulysses22	30/30	30/30
gr202	**30/30**	29/30
tsp225	00/30	**02/30**
a280	00/30	**19/30**
pcb442	00/30	**30/30**
gr666	**30/30**	03/30

The Opt/Runs means how many times the algorithm reach the known optimum solution from the total number of runs. For example, 19/30 represents the algorithm reach the known optimum solution for 19 times from 30 runs. In the table, EDFA performed slightly better for gr202 and significantly better for gr666 instance than Memetic algorithm. But, for three TSP instances, tsp225, a280 and pcb442, EDFA performed much worse, where it never reached the known optimum solution. Even, according to Table 3, it reached solutions with low accuracies, only 88%, from the known optimum solutions. This shows that EDFA can be trapped into local optimum solutions for some instances.

3.6 Some Ideas to Improve EDFA

In order to solve TSP, we can improve EDFA by giving direction to the fireflies. Direction can be formed by dividing the TSP instance problem into multiple sub partitions. Each sub partitions can be considered as the direction. We can divide the problem using various ways. The easiest way to divide is based on the coordinates, by X or Y axis. Figure 7 showed TSP instance divided into two sub partitions by Y axis. Another way is to use K-mean to divides the problem into multiple sub-partitions.

Then solves each sub partition separately, and finally use the partial solution to produce a complete tour.

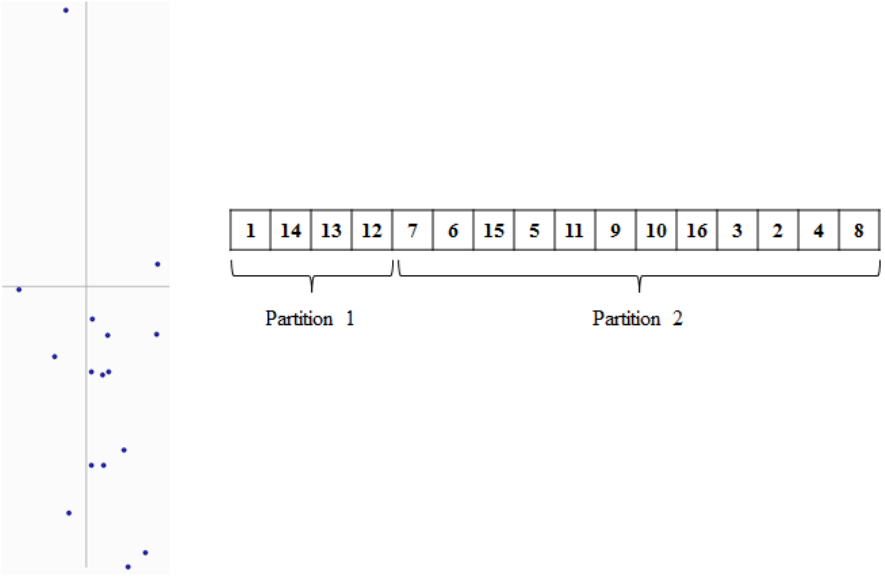

Fig. 7. TSP instance ulysses16 divide into two sub partitions by Y axis

In EDFA, the partition has a role in the movement of fireflies ïïWhen a firefly *i* is attracted to another brighter (more attractive) firefly *j*, it calculates the long of the steps should be taken for each partition. Here, a partition can be seen as a dimension in a vector. If there are two partitions, then a firefly can, for example, do a long movement for the first partition and short movement for the second one. Hence, the firefly seems to have a direction of movement. However, the number of partition can be more than two. If a TSP instance has thousands of cities and the cities are distributed into 10 clusters, then the instance is better to divide into 10 partitions. Hence, a firefly can be seen as an object that moves in 10 dimensional vector spaces with a long of step and a direction.

EDFA could be combined with other methods such as Greedy and Lin Kernighan (LK). Greedy works simply by growing Hamiltonian cycle in the graph in which it is necessary that each vertex should be visited only once. Greedy can create this Hamiltonian cycle by first picking the shortest edge and then add this shortest edge to the tour until all the cities are included in the tour. Greedy can repeat this whole procedure until all cities be the part of the Hamiltonian cycle. Greedy can create better tour for initialization of fireflies. LK is one of the best algorithms of TSP which generates an optimal solution for the TSP. LK is the most powerful heuristic. It works as follows. First, it removes an edge from the current tour, and tries to add edges with the other cities in such a way that will create the new shortest tour. In EDFA, LK can also be used to update each partial solution (partition).

4 Conclusion

The proposed method, EDFA, has been successfully implemented to solve TSP. The simulation results indicate that EDFA performs very well for some TSP instances compare to Memetic algorithm. But, it can be trapped into local optimum solutions for some instances since it does not have a direction to do a movement. For improvement, it could be combined with other techniques such as greedy search for the initialization of firefly population or Lin Kernighan algorithm to update each partial solution in a partition and to further improvement of the tour.

Acknowledgments. We would like to thank to Dr Xin-She Yang in Cambridge University for the suggestions and comments on the initial manuscript and our colleagues in Telkom Institute of Technology (IT Telkom) for the advices and motivations.

References

1. Luciana, B., França, P.M., Pablo, M.: A New Memetic Algorithm for the Asymmetric Traveling Salesman Problem. Journal of Heuristics 10(5), 483–506 (2004)
2. Lukasik, S., Żak, S.: Firefly algorithm for Continuous Constrained Optimisation Tasks. Systems Research Institute, Polish. Academy of Sciences, pp. 1–10 (2010)
3. Niasar, N.S., Shanbezade, J., Perdam, M.M.: Discrete Fuzzy Particle Swarm Optimization for Solving Travelling Salesman Problem. In: Proceedingsof International Conference on Information and Financial Engineering, pp. 162–165 (2009)
4. Pilat, M.L., White, T.: Using Genetic Algorithms to Optimize ACS-TSP. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) Ant Algorithms 2002. LNCS, vol. 2463, pp. 12–14. Springer, Heidelberg (2002)
5. Sayadi, M.K.: A Discrete Meta-Heuristic With Local Search for Makespan Minimization in Permutation Flow Shop Scheduling Problems. International Journal of Industrial Engineering Computation 1(1), 1–10 (2010)
6. TSPLIB95: Ruprecht - Karls - Universitat Heildelberg (2011), `http://www.iwr.uini-heidelberg.de/groups/comopt/software/TSPLIB95/`
7. Yang, X.S.: Nature-inspired Metaheuristic Algorithm. Luniver Press (2008)
8. Yang, X.S.: Firefly Algorithms for Multimodal Optimization. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009)
9. Yang, X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications. Wiley, Chichester (2010)
10. Tsai, H., Yang, J., Tsai, Y., Kao, C.: An Evolutionary Algorithm for Large Traveling Salesman Problems. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 1718–1729 (2004)
11. Wong, L., Chong, C.S.: An Efficient Bee Colony Optimization Algorithm for Travelling Salesman Problem Using Frequency-Based Pruning. In: Proceedings of 7th IEEE International Conference on Industrial Informatics, pp. 775–782 (2009)