International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

http://ijpsat.ijsht-journals.org
Vol. 28 No. 2 September 2021, pp.672-679

Selected Soft Computing Algorithms For Solving Travelling Salesman Problem

Adedeji Oluyinka Titilayo[1]*, Alade Oluwaseun Modupe[2*], Makinde Bukola Oyeladun[3*], Jooda Janet Olubunmi[4*], Adewale Joseph Adekunle[5*], Oyeleye Taye E[6*]

[1]Department of information Systems, Faculty of Computing and Informatics, Ladoke Akintola University of Technology (LAUTECH), Nigeria

[2]Department of Cyber Security Science, Ladoke Akintola University of Technology, Ogbomoso, Nigeria

[3]Department of Computer Science, Faculty of Communication, Information and Technology (CIT), Osun State College of Technology, Esaoke

[4]Department of Computer Engineering, Igbajo Polytechnic, Igbajo, Osun State. Nigeria.

[5]Department of Multimedia Technology, Osun State College of Technology, Esaoke.

[6]Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria

***Abstract*— Traveling Salesman Problem (often called TSP) is a classic algorithmic problem in the field of computer science and operations research. It is focused on optimization. In this context, better solution often means a solution that is cheaper, shorter, or faster. TSP is a mathematical problem. It is most easily expressed as a graph describing the locations of a set of nodes. Given a set of cities and distance between every pair of cities, the problem of Traveling Salesman Problem is to find the shortest possible route that visits every city exactly once and returns to the starting point. The aim of this project is to adapt Bat, Bee, Firefly, and Flower pollination algorithms, implement and evaluate the selected algorithms for solving Travelling Salesman Problem.**

Keywords— Optimization, BAT, BEE, TSP, Transportation

I. Introduction

Travelling Salesman Problem (TSP) is a Non-deterministic Polynomial (NP) problem that is easy to describe but difficult to solve. In TSP a salesman has to visit a number of cities such that it must visit every city only once. The problem is to find the tour the salesman should follow such that the path length should be minimum as compared to other possible routes (Charu Vasishta, Preethi Jose Dolly, 2015). This problem is known to be Non-deterministic Polynomial (NP)-hard, and cannot be solved exactly in polynomial time. TSP is solved very easily when there are lesser number of cities, but as the number of cities increases it becomes very hard to solve, as large amount of computation time is required. Many exact and heuristic algorithms have been developed in the field of operations research (OR) to solve this problem. Another approach is to use genetic algorithm to solve TSP because of its robustness and flexibility. Some typical applications of TSP include vehicle routing, computer wiring, cutting wallpaper and job sequencing. The main application in statistics is combinatorial data analysis, e.g., reordering rows and columns of data matrices or identifying clusters. (Saloni Gupta & Poonam Panwar, 2013).

Traditional method such as brute force search is unfeasible to be applied to solve TSP. The brute force method for solving the TSP checks every possible route and takes the shortest one as the answer. Unfortunately, the number of routes grows exceptionally fast. If this problem only needed to find the route with 10 cities, it would be manageable. If there were 20 or more cities, the number of possible routes would have longer digits. Obviously, this method quickly becomes unfeasible. Other problem such as premature convergence and getting stuck in local optima would be given proper attention.

Therefore, this project implements bat algorithm, bee algorithm, firefly algorithm and flower pollination algorithm to solve TSP, considering their potential in various fields of optimization problems.

II. Literature Review

Travelling Salesman problem (TSP) is one of the most popular problems from the Non-deterministic Polynomial (NP) set, it is also one of the hardest too. The solution to this problem enjoys wide applicability in a variety of practical fields. Thus, it highly raises the need for an efficient solution for this Non-deterministic Polynomial (NP) Hard problem. However even the most common deterministic solutions to the problem are known to possess an exponential time complexity. There have been many efforts in the past to provide time efficient solutions for the problem, both exact and approximate. This project surveys on some of these deterministic and non-deterministic efforts. It presents and analyzes a greedy non−deterministic algorithm to solve the travelling salesman problem and relates the Travelling Salesman problem with the Hamiltonian circuit problem in order to demonstrate the existing property of reducibility between the two problems and the its advantages in providing a solution to the TSP. (Sanchit Goyal, 2010)

TSP is a useful problem in routing problems e.g. in a transportation system. There are different approaches for solving TSP. Solving TSP was an interesting problem during recent decades. Almost every new approach for solving engineering and optimization problems has been tested on TSP as a general test bench. First steps in solving TSP were classical methods. These methods consist of heuristic and exact methods. Heuristic methods like cutting planes and branch and bound (Padherg & Rinaldi, 1987), can only optimally solve small problems whereas the heuristic methods, such as 2-opt (Lin & Kernighan, 1973), 3-opt, Markov chain (Martin et al., 1991), simulated annealing (Kirkpatrick et al., 1983) and tabu search are good for large problems.

Figure 1: Shortest distance around cities. (Michael Schirber, 2017)

III. Bat Algorithm

Bat algorithm was developed by Xin-She Yang in 2010. The algorithm exploits the so-called echolocation of the bats. The bats use sonar echoes to detect and avoid obstacles. It is generally known that sound pulses are transformed into a frequency which reflects from obstacles. The bats navigate by using the time delay from emission to reflection. They typically emit short, loud sound impulses.

IV. Bee Algorithm

The BCO was inspired by bees behavior in the nature. The basic idea behind the BCO is to create the multi agent system (colony of artificial bees) capable to successfully solve difficult combinatorial optimization problems. The Artificial Bee Colonies (ABC) is another novel optimization algorithm that comes under Swarm Intelligence. ABC algorithm is inspired by social behavior of natural bees. (Karaboga and Basturk, 2007).

V. Firefly Algorithm

Firefly algorithm is inspired by the social behavior of fireflies. Most of the fireflies produce short and rhythmic flashes and have different flashing behavior. Fireflies use these flashes for communication and attracting the potential prey. The swarm of fireflies will move to brighter and more attractive locations by the flashing light intensity that is associated with the objective function of problems considered, in order to obtain efficient optimal solutions. One major improvement is the firefly algorithm (FA) which was based on the flashing characteristics of tropical fireflies. The attraction behavior, light intensity coding, and distance dependence provides a surprising capability to enable firefly algorithm to handle nonlinear, multimodal optimization problems efficiently (Xin She Yang, 2008).

VI. Flower Pollination Algorithm

In nature, the main purpose of the flowers is reproduction via pollination. Flower pollination is related to the transfer of pollen, which is done by pollinators such as insects, birds, bats, other animals or wind. Some flower types have special pollinators for successful pollination. The four rules of pollination have been formulated based on the inspiration from flowering plants and they form the main updating equations of the flower pollination algorithm, the main actors of performing such transfer are birds, bats, insects, and other animals. There exist some flowers and insects that have made what we can call a flower-pollinator partnership. These flowers can only attract the birds that are involved in that partnership, and these insects are considered the main pollinators for these flowers (Glover, 2007).

VII. Problem Description

Traditional method such as brute force search is unfeasible to be applied to solve TSP. The brute force method for solving the TSP checks every possible route and takes the shortest one as the answer. Unfortunately, the number of routes grows exceptionally fast. If this problem only needed to find the route with 10 cities, it would be manageable. If there were 20 or more cities, the number of possible routes would have longer digits. Obviously, this method quickly becomes unfeasible. Other problem such as premature convergence and getting stuck in local optima would be given proper attention.

Therefore, this project implements bat algorithm, bee algorithm, firefly algorithm and flower pollination algorithm to solve TSP, considering their potential in various fields of optimization problems.In this paper, Bat, Bee, Firefly and Flower Pollination algorithm were adapted and implemented for solving Travelling Salesman Problem and to carry out performance evaluation of the selected algorithms for solving Travelling Salesman Problem.

VIII. Methodology

Finding a solution to Travelling Salesman Problem requires that we setup the bat, bee, firefly and flower pollination algorithms in a specialized way. The following parameters were used to carry out this project.

BAT

Bat algorithm was guided by the following parameters:

I. the population size: 30
II. loudness: 0.25
III. pulse rate: 0.1
IV. search variable dimension: 50
V. maximum iteration: 100

BEE

Bee algorithm was guided by the following parameters:

i. number of scout bees: 30

ii. maximum iteration: 100

FIREFLY

Firefly algorithm was guided by the following parameters:

i. maximum iteration: 100

ii. number of fireflies: 100

iii. attraction coefficient: 0.2

iv. mutation coefficient: 0.2

v. light coefficient: 1

FLOWER POLLINATION

Flower Pollination algorithm was guided by the following parameters:

i. switch probability: 0.8

ii. population size: 100

iii. maximum iteration: 100

In this paper, the distances between the cities are stored in a symmetric matrix as shown in figure 2. Here in this paper, there are 20 cities to be visited by the salesman.

```
=====================================================================
Distance Matrix
=====================================================================
      1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20

 1:   0  39  36  46  45  40  26  26  36  23  16  22  13  44  38  20  37  30  37  18
 2:  39   0  30  27  26  19  14  27  12  33  19  28  33  40  33  47  39  25  34  33
 3:  36  30   0  44  30  19  24  25  28  10  15  33  23  38  30  48  31  47  32  16
 4:  46  27  44   0  28  40  42  20  25  47  47  15  36  13  34  37  48  33  33  47
 5:  45  26  30  28   0  38  44  37  29  16  10  11  49  36  31  43  20  37  25  23
 6:  40  19  19  40  38   0  17  16  19  26  21  35  46  35  12  48  47  24  43  12
 7:  26  14  24  42  44  17   0  15  32  30  38  16  36  19  15  19  18  32  49  32
 8:  26  27  25  20  37  16  15   0  35  43  41  29  24  15  17  43  29  48  16  46
 9:  36  12  28  25  29  19  32  35   0  36  20  46  35  35  26  19  35  14  16  25
10:  23  33  10  47  16  26  30  43  36   0  36  19  26  11  27  19  31  45  15  21
11:  16  19  15  47  10  21  38  41  20  36   0  26  31  29  15  14  25  48  39  39
12:  22  28  33  15  11  35  16  29  46  19  26   0  36  35  14  27  38  37  15  10
13:  13  33  23  36  49  46  36  24  35  26  31  36   0  36  25  46  38  41  16  23
14:  44  40  38  13  36  35  19  15  35  11  29  35  36   0  16  28  39  15  12  14
15:  38  33  30  34  31  12  15  17  26  27  15  14  25  16   0  48  30  44  14  31
16:  20  47  48  37  43  48  19  43  19  19  14  27  46  28  48   0  20  42  19  20
17:  37  39  31  48  20  47  18  29  35  31  25  38  38  39  30  20   0  48  29  15
18:  30  25  47  33  37  24  32  48  14  45  48  37  41  15  44  42  48   0  27  36
19:  37  34  32  33  25  43  49  16  16  15  39  15  16  12  14  19  29  27   0  22
20:  18  33  16  47  23  12  32  46  25  21  39  10  23  14  31  20  15  36  22   0
=====================================================================
```

Figure 2: Distance Matrix For 20 Cities

IX. RESULT ANALYSIS

In other to check that the proposed algorithm gives optimal result, each of the algorithms (Bat, Bee, Firefly and Flower Pollination) were applied to 20 cities respectively, setting iteration limit to 100 for 5 successive runs.

After using Firefly algorithm on 20 cities for the first run, the best tour length= 259.5799 and the best simulation time= 15.66618 as shown in Figure 3.1 and Figure 3.2.

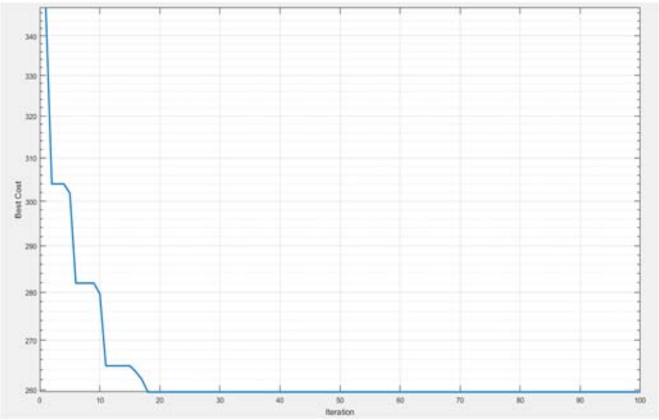

Figure 3.1: Convergence Curve Graph Using Firefly Algorithm

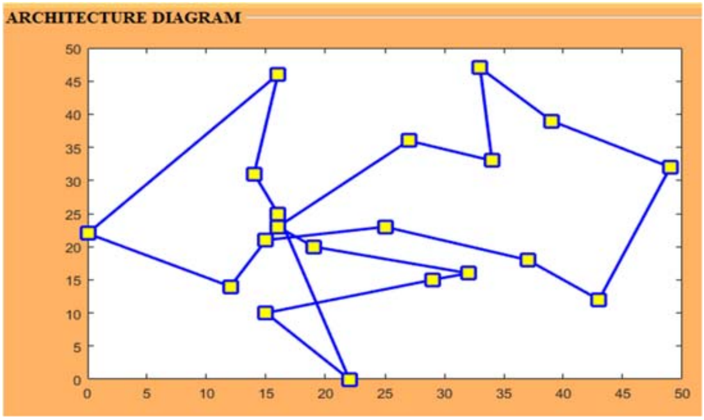

Figure 3.2: Architecture Diagram Showing City Nodes Using Firefly Algorithm

After using Bee algorithm on 20 cities for the first run, the best tour length= 295.0347 and the best simulation time= 5.141175 as shown in Figure 3.3 and Figure 3.4.

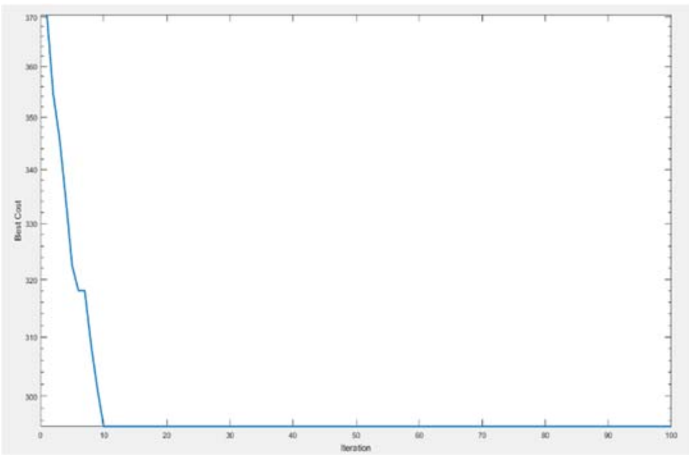

Figure 3.3: Convergence Curve Graph Using Bee Algorithm

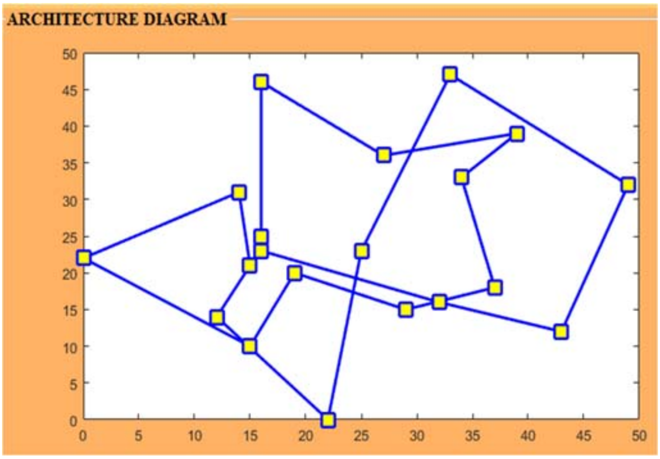

Figure 3.4: Architecture Diagram Showing City Nodes Using Bee Algorithm

After using Bat algorithm on 20 cities for the first run, the best tour length= 299.0778 and the best simulation time= 3.152378 as shown in Figure 3.5 and Figure 3.6.

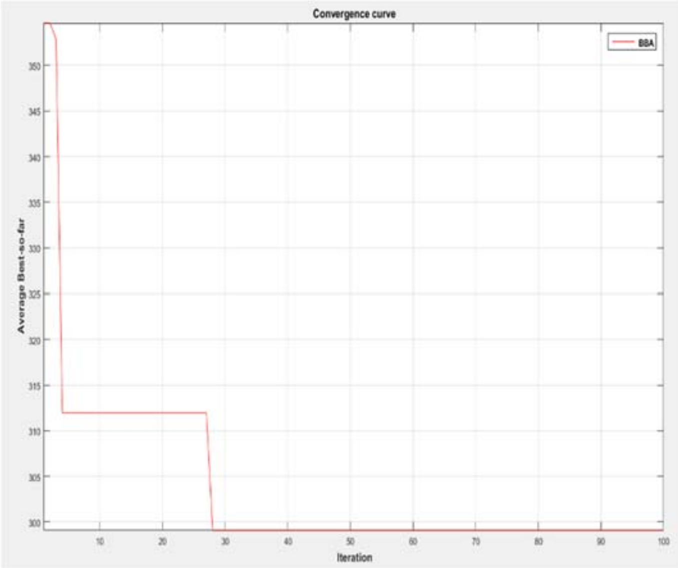

Figure 3.5: Convergence Curve Graph Using Bat Algorithm

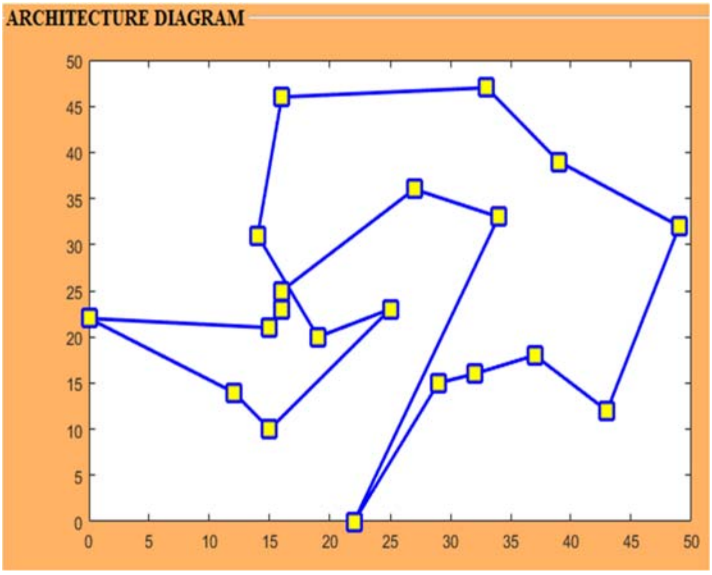

Figure 3.6: Architecture Diagram Showing City Nodes Using Bat Algorithm

After using Flower Pollination algorithm on 20 cities for the first run, the best tour length= 299.0778 and the best simulation time= 3.152378 as shown in Figure 5.7 and Figure 5.8.

Table 1 : Comparison of Algorithms Based on Tour Length and Simulation Time

Experimental Run	**FIREFLY ALGORITHM**		**BEE ALGORITHM**		**BAT ALGORITHM**		**FLOWER POLLINATION ALGORITHM**	
	Best Tour Length(m)	Best Simulation Time(s)	Best Tour Length(m)	Best Simulation Time(s)	Best Tour Length(m)	Best Simulation Time(s)	Best Tour Length(m)	Best Simulation Time(s)
FIRST	259.5799	15.66618	295.0347	5.141175	299.0778	3.152378	283.9801	1.710143
SECOND	239.9813	27.79587	307.984	4.23078	310.2788	3.233572	256.1302	1.65762
THIRD	247.5026	16.38627	286.0819	4.158456	298.8589	3.112403	268.7823	1.693628
FOURTH	258.8089	20.28618	274.0186	4.207436	315.0342	3.227888	292.2728	1.689731
FIFTH	241.0052	15.3885	277.9475	4.210528	299.0778	3.117539	235.3088	1.799142
AVERAGE	249.3756	22.2378	288.2133	4.3897	304.4655	3.1688	267.2948	1.7089

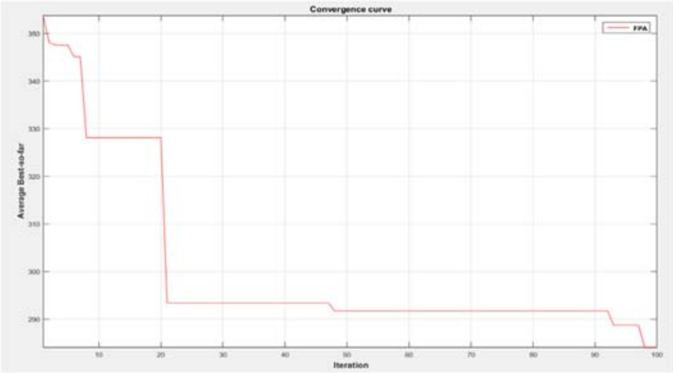

Figure 3.7: Convergence Curve Graph Using Flower Pollination Algorithm

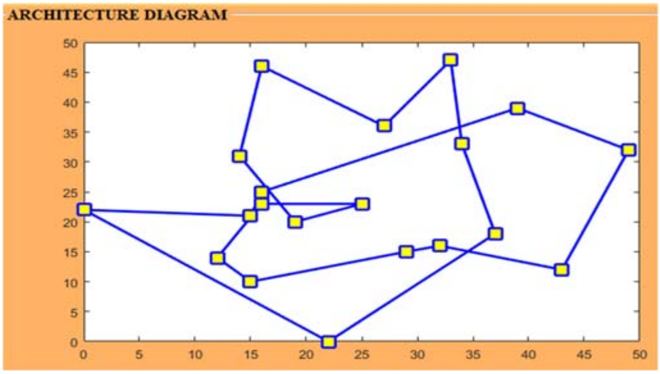

Figure 3.8: Architecture Diagram Showing City Nodes Using Flower Pollination Algorithm

Based on the above comparison table for test data, Flower Pollination Algorithm yields comparatively better tour length of 267.2948km in very less simulation time when compared to others, although Firefly Algorithm yielded the best tour length of 249.3756km by a slight margin but takes longer time to simulate than all other selected algorithms.

Performance Metrics Result

The result of the software complexity metrics is given in Table:

Table 2: Software Complexity Metrics for each algorithm

Metrics	**BAT**	**BEE**	**FIREFLY**	**FLOWER POLLINATION**
Vocabulary(n)	104	102	100	163
Lines of Code (LOC)	63	124	128	184
Cyclomatic Complexity (G)	7	13	9	14
Calculated Program Length (N_h)	652.6656	643.7878	624.1677	1226.3293
Maintainability Index (M.I)	78.1204	65.8716	66.3802	56.8105

From the above table, Firefly algorithm had the best and Vocabulary(n) with a unit of 100 and the best Calculated Program Length (N_h) with a unit of 624.1677. However, Bat algorithm had a Cyclomatic Complexity of 7 while Flower Pollination algorithm had a Maintainability index of 56.8105. It can be stated that Firefly algorithm performed better in term of Software Complexity.

X. Conclusion And Future Work

In this paper, Bat, Bee, Flower Pollination and Firefly Algorithms were adapted and implemented for solving Travelling Salesman Problem. Th^{2}e experimental results obtained on standard TSP instances showed that Firefly Algorithm (FA) provides better results than Bat, Bee and Flower Pollination Algorithms in most of the instances. Future works can be tailored towards hybridization of two algorithms to solve Travelling Salesman Problem and observe whether there is any improvement to the results presented to this work.

REFERENCES

[1] Charu Vasishta, Preethi Jose Dolly (2015): Solving Travelling Salesman Problem using Genetic Algorithm by Combining Greedy Approach and Ordered Cross Over. *International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 6.*

[2] Dervis Karaboga, Bahriye Basturk (2007): A powerful and efficient algorithm for numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm J Glob Optim, Volume 39, Issue 3, pp 459–471

[3] Glover B.J (2007): Understanding Flowers and Flowering: An integrated approach. Understanding Flowers and Flowering: An integrated approach. 1-256. 10.1093/acprof:oso/9780198565970.001.0001.

[4] Saloni Gupta and Poonam Panwar (2013): Solving Travelling Salesman Problem Using Genetic Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering.*

[5] Yang, X. S., (2010). A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. Cruz, C.; Gonz´alez, J. R.; Pelta, D. A.; Terrazas, G), Studies in Computational Intelligence Vol. 284, Springer Berlin, pp. 65–74.

[6] Yang X-S (2008) Nature-inspired metaheuristic algorithm. Luniver Press, Beckington.

[7] Yang X-S (2009) Firefly algorithms for multimodal optimization, In: Stochastic algorithms: foundations and applications, SAGA, Lecture Notes in Computer Sciences, 5792, 169, 178.