

# Selected Soft Computing Algorithms for Job Shop Problem (JSP)

Adedeji Oluyinka Titilayo<sup>\*1</sup>  
Department of Information  
System Science, Ladoke  
Akintola University of  
Technology, Ogbomoso,  
Nigeria  
otadedeji@lautech.edu.ng

Alade Oluwaseun Modupe<sup>2\*</sup>  
Department of Cyber Security  
Science, Ladoke Akintola  
University of Technology,  
Ogbomoso, Nigeria  
oalade75@lautech.edu.ng

Makinde Bukola  
Oyeladun<sup>3\*</sup>  
Department of Computer  
Science, Osun State College of  
Technology, Esa-Oke. Nigeria  
bukolamakinde22@gmail.com

OYELEYE Taye E  
Department of Computer  
Science and Engineering,  
Ladoke Akintola University of  
Technology, Ogbomoso,  
Nigeria

(\*Corresponding author's e-mail: otadedeji@lautech.edu.ng, oalade75@lautech.edu.ng,  
bukolamakinde22@gmail.com)

---

**Abstract:** Job Shop Problem (JSP) is an optimization problem in computer science and operations research in which jobs are assigned to resources at particular times. Each operation has a specific machine that it needs to be processed on and only one operation in a job can be processed at a given time. This problem is one of the best known combinatorial optimization problems. The aim of this project is to adapt Bat, Bee, Firefly, and Flower pollination algorithms, implement and evaluate the developed algorithms for solving Job Shop Problem.

**Keywords:** scheduling. Optimization, job shop problem, BAT, BEE)

---

## 1. INTRODUCTION

Scheduling is the allocation of shared resources over time to competing activities. It has been the subject of a significant amount of literature in the operations research field. Emphasis has been on investigating machine scheduling problems where jobs represent activities and machines represent resources; each machine can process at most one job at a time practical and varied. They arise in diverse areas such as flexible manufacturing systems, production planning, computer design, logistics, communication, etc. A scheduling problem is to find sequences of jobs on given machines with the objective of minimizing some function of the job completion times.

In a simpler version of this problem, flow shop scheduling, all jobs pass through all machines in the same order. A more complex case is represented by a job shop scheduling problem where machine orderings can be different for each job. Job shop problem (JSP) is one of the hardest combinatorial optimization problems. It belongs to the class of Non-deterministic Polynomial (NP) hard problems, consequently there are no known algorithms guaranteed to give an optimal solution and run in polynomial time. That means, classical optimization methods (branch and bound method, dynamic programming) can be used only for small scale tasks. (Binato, Hery, Loewenstern and Resende, 2002).

## 2. THEORETICAL BACKGROUND

### Job Shop Problem (JSP)

JSP is a static optimization problem, since all information about the production program is known in advance. General job shop problem is the probably most studied one by academic research during the last three decades and is notoriously difficult problem to solve. The JSP is a Non-deterministic Polynomial (NP) hard problem and among those optimization problems, it is one of the least tractable known problem (Garey and Johnson, 1979). It is purely deterministic, since processing time and constraints are fixed and no stochastic events occur.

JSP also illustrates some of the demands required by a wide array of real-world problems. In a shop floor, machines process jobs and each job contain a certain number of operations. Each operation has its own processing time and has to be processed on a dedicated machine. Each job has its own machine order and no relation exists between machine orders of any two jobs. For each job, the machine order of operations is prescribed and is known as technological production recipe or technological constrain, which are static to a problem instance. (Garey and Johnson, 1979).

Operations to be processed on one machine form an operation sequence for this machine. For a given problem, an operation sequence for each machine is called a schedule. Since each operation sequence can be permuted independently of operation sequences of other machines, the problem with  $n$  jobs and  $m$  machines can have a maximum of different solutions. The completion time of all jobs is known as makespan. The objective is to find a feasible schedule with minimum makespan. Feasible schedules are obtained by permuting the processing order of operations on machines without violating the technological constraints. (El-Bouri, Azizi and Zolfaghari, 2007)

Makoto and Hiroshi considered the JSP problem to minimize the total weighted tardiness with job-specific due dates and delay penalties, and a heuristic algorithm based on the tree search procedure was developed for solving the problem. Gomes and Barbosa presented an integer linear programming model to schedule flexible job shop, which considered job re-circulation and parallel homogeneous machines Loukit and Jacques dealt with a production scheduling problem in a flexible job shop with particular constraints-batch production.

An example of two jobs to be performed three machines (2x3) job shop problem is illustrated in Table 1 In this problem, each job requires three operations to be processed on a pre-defined machine sequence. The first job ( $J_1$ ) need to be initially operated on the machine  $M_1$  for 5time units and then sequentially processed on  $M_2$ and  $M_3$  for 4 and 9 time units, respectively. Likewise, the second job ( $J_2$ ) has to be initially performed on  $M_3$  for 5 time units and sequentially followed by  $M_1$  and  $M_2$  for 6 and 7 time units, respectively. The design task for solving JSP is to search for the best schedule(s) for operating all pre-defined jobs in order to optimize either single or multiple scheduling objectives, which is used for identifying a goodness of schedule such as the minimization of the makespan ( $C_{max}$ ) (Ge HW, Sun, Liang, Qian, 2009).

**Table1** An example of 2-jobs 3-machines scheduling problem with processing times.

(Ge HW, Sun , Liang, Qian, 2009)

Job ( $J$ )	Operation ( $O_{jk}$ )	Time ( $t_{jk}$ )	Machine ( $M_j$ )		
			$M_1$	$M_2$	$M_3$
$J_1$	$O_{11}$	5	5	-	-
	$O_{12}$	4	-	4	-
	$O_{13}$	9	-	-	9
$J_2$	$O_{23}$	5	-	-	5
	$O_{21}$	6	6	-	-
	$O_{22}$	7	-	7	-

## I. 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.

## II. 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).

## III. 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).

## IV. 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).

## 3. RELATED WORKS

Several works have been done in recent years to solve the Job Shop Problem. Xueni and Henry (Xueni Qiu and Henry Lau, 2012) proposed a new hybrid algorithm based on Particle Swarm Optimization (PSO) and Artificial Immune Systems (AIS) theories of clonal selection and immune network to solve the job shop scheduling problem, which is a classical combinatorial optimization problem. Experimental results demonstrated that the algorithm was competitive among other methods and optimal solutions were obtained within a reasonable computation time, especially for small size problems. However, there were occurrence of unexpected events and disturbances in the scheduling process.

Chaudry (Chaudry, 2012) presented a modified permutation chromosome representation for schedules with alternative machines for given operations. Although all the alternatives were included in the chromosome, the second and subsequent appearances of an operation in the chromosome

did not contribute to the overall calculation of the fitness value. However, all occurrences were included in the crossover and mutation operations, allowing for alternatives.

This chromosome representation did not represent a customization of the Genetic Algorithm (GA) method to accommodate a particular problem. Instead, it was a generalization of the existing representation, which in turn became a reduced form of this generalized form. However, the second chromosome representation also produced the same results as the modified permutation representation.

Beck and Wilson (2007) addressed job shop scheduling when the durations of the activities were independent random variables. A theoretical framework was created to formally define this problem and to prove the soundness of two algorithm components: Monte Carlo simulation to find upper bounds on the probabilistic makespan of a solution and a partial solution; and a carefully defined deterministic JSP whose optimal makespan was a lower bound on the probabilistic makespan of the corresponding probabilistic JSP. Then used these two components together with either constraint programming or Tabu search to define a number of algorithms to solve probabilistic JSPs, introduced three solution approaches: a branch-and-bound technique using Monte Carlo simulation to evaluate partial solutions; an iterative deterministic search using Monte Carlo simulation to evaluate the solutions from a series of increasingly less constrained problems based on a parameterizable lower bound; and a number of deterministic filtering algorithms which generate a sequence of solutions to a deterministic JSP, each of which was then simulated using Monte Carlo simulation. Empirical evaluation demonstrated that the branch-and-bound technique was only able to find approximately optimal solutions for very small problem instances. The iterative deterministic search performed as well as, or better than, the branch-and-bound approach for all problem sizes. However, for medium and large instances, the deterministic filtering techniques performed much more strongly while providing no optimality guarantees.

The scheduling of jobs to machines is a very challenging problem with several constraints that have to be explored in different ways. The use of certain heuristics and exact methods have been used to solve JSP such as Tabu Search. Although Tabu Search has gained popularity in recent years in terms of space i.e. space complexity, however, it is hard to understand the concept of k-insertion and implementing the algorithms. Other problems such as premature convergence and getting stuck in local optima would be given proper attention.

Therefore, Bat, Bee, Firefly and Flower Pollination algorithms were adapted and implemented for solving Job Shop Problem and to carry out performance evaluation of selected algorithms for solving Job Shop Problem.

#### 4. METHODOLOGY

Finding a solution to the Job Shop Problem requires that we set up 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. population size: 100
- 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 co-efficient: 0.2
- iv. mutation co-efficient: 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 sequences for each job are stored in a symmetric matrix, as shown in the figure 2. Here in this paper are 6 jobs to be carried out on 6 machines.

$$T = \begin{bmatrix} 1 & 3 & 6 & 7 & 3 & 6 \\ 8 & 5 & 10 & 10 & 10 & 4 \\ 5 & 4 & 8 & 9 & 1 & 7 \\ 5 & 5 & 5 & 3 & 8 & 9 \\ 9 & 3 & 5 & 4 & 3 & 1 \\ 3 & 3 & 9 & 10 & 4 & 1 \end{bmatrix};$$

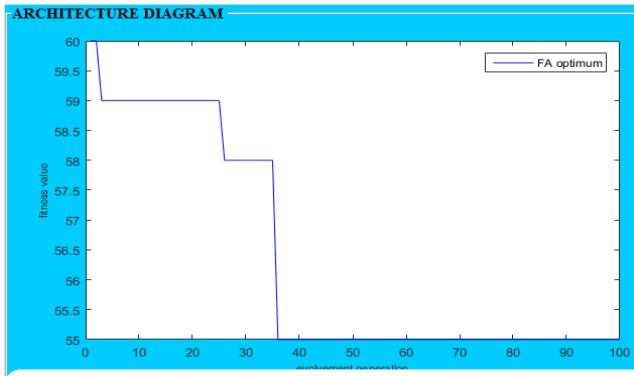
$$js = \begin{bmatrix} 3 & 1 & 2 & 4 & 6 & 5 \\ 2 & 3 & 5 & 6 & 1 & 4 \\ 3 & 4 & 6 & 1 & 2 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 3 & 2 & 5 & 6 & 1 & 4 \\ 2 & 4 & 6 & 1 & 5 & 3 \end{bmatrix};$$

Figure 2: Data for 6 jobs using 6 machines

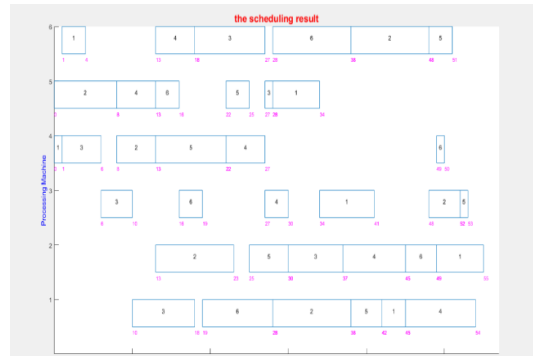
#### 5. RESULTS AND DISCUSSION

In other to check that the proposed algorithms give optimal result, each of the algorithms (Bat, Bee, Firefly and Flower Pollination) were applied to 6 numbers of jobs using 6 machines respectively, setting iteration limit to 100 for 5 successive runs.

After using Firefly algorithm on 6 jobs and 6 machines for the first run, the best processing time = 55s as shown in Figure 3.1 and Figure 3.2.

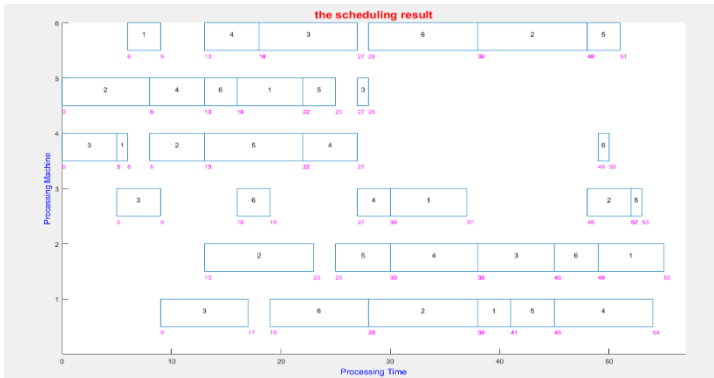


**Figure3.1:** Architecture Diagram showing fitness value using Firefly algorithm



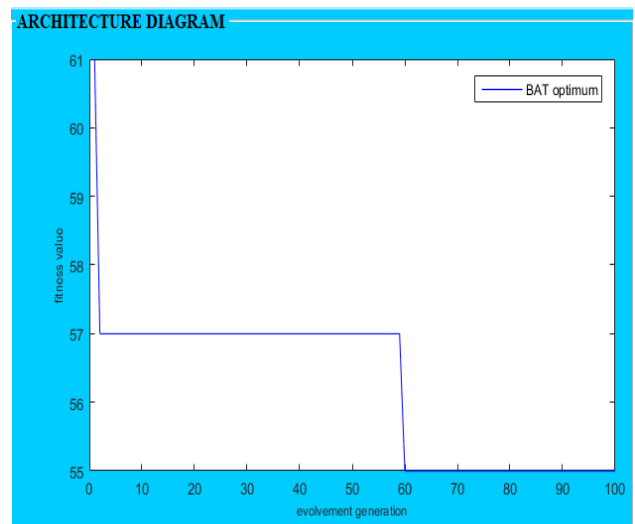
**Figure3.4:** Scheduling Result Showing Time Sequence using Bee algorithm

After using Bat algorithm on 6 jobs and 6 machines for the first run, the best processing time = 55s as shown in Figure 3.5 and Figure 3.6.

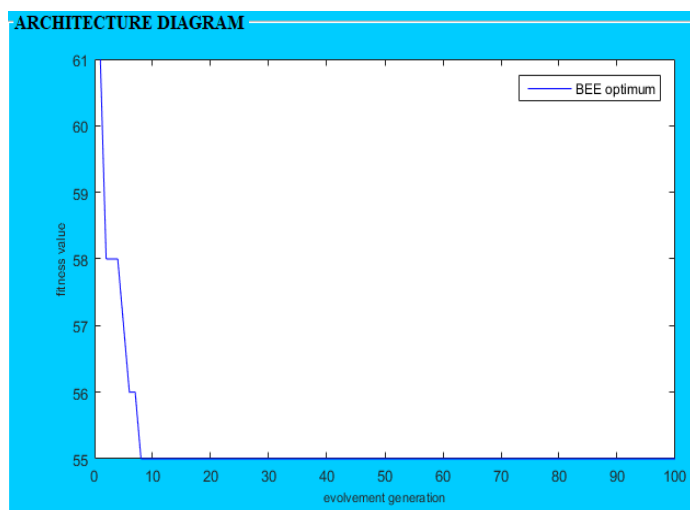


**Figure3.2:** Scheduling Result Showing Time Sequence using Firefly algorithm

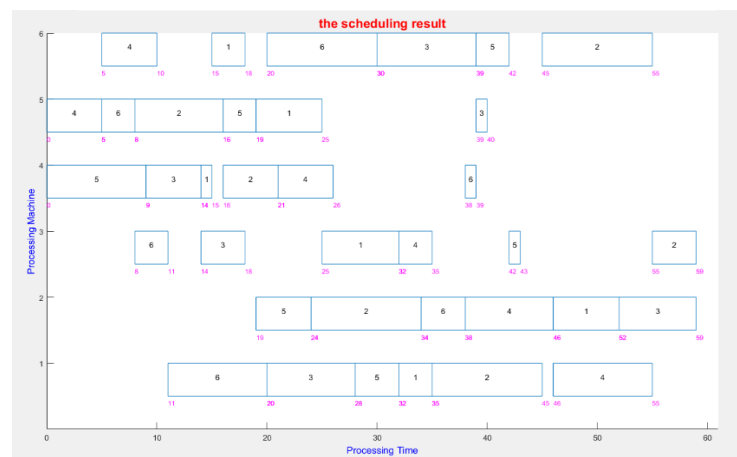
After using Bee algorithm on 6 jobs and 6 machines for the first run, the best processing time = 55s as shown in Figure 3.3 and Figure 3.4.



**Figure3.5:** Architecture Diagram showing fitness value using Bat algorithm



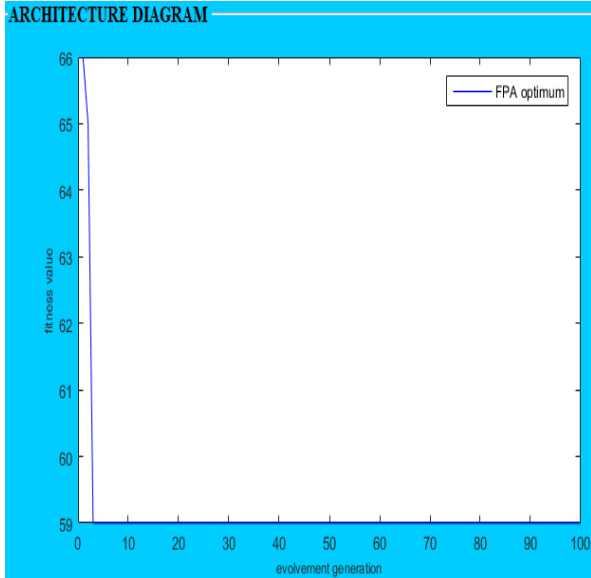
**Figure3.3:** Architecture Diagram showing fitness value using Bee algorithm



**Figure3.6:** Scheduling Result Showing Time Sequence using Bat algorithm

After using Flower Pollination algorithm on 6 jobs and 6 machines for the first run, the best processing time = 59s as shown in Figure 3.7 and Figure 3.8.

**Table 2:** Comparisons of Algorithms Based on Processing Time

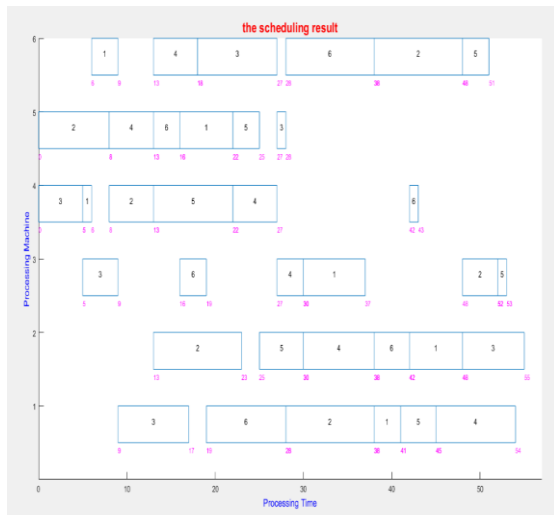


**Figure3.7:** Architecture Diagram showing fitness value using Flower Pollination algorithm

Experimental Run	FIREF	BEE	BAT	FLOWER POLLINATION
	Best Processing Time(s)	Best Processing Time(s)	Best Processing Time(s)	Best Processing Time(s)
FIRST	55	55	55	59
SECOND	57	57	55	55
THIRD	55	55	58	55
FOURTH	55	57	55	55
FIFTH	57	55	55	55
AVERAGE	55.8	55.8	55.6	55.8

Based on the above comparison table for test data, Bat Algorithm yielded comparatively the best processing time of 55.6 seconds when compared to other selected algorithms.

**Table 3:** Software Complexity Metrics for Each Algorithm



**Figure3.8:** Scheduling Result Showing Time Sequence using Flower Pollination algorithm

### Performance Metrics Result

The result of the software complexity metrics is given in Table 3:

Metrics	BAT	BEE	FIREFLY	FLOWER PA
Vocabulary(n)	82	89	102	148
Lines of Code (LOC)	63	89	92	91
Cyclomatic Complexity(G)	11	11	9	8
Calculated Program Length (N <sub>h</sub> )	480.3865	540.9707	640.1314	1018.0621
Maintainability Index (M.I)	78.4369	72.4140	71.6290	70.0986

From the above table, Bat algorithm had the best Vocabulary(n) with a unit of 82 and the best Calculated Program Length (N<sub>h</sub>) with a unit of 480.3865. However, Flower Pollination algorithm had a Cyclomatic Complexity of 8, Maintainability Index of 70.0986. It can be stated that Bat and Flower Pollination algorithms performed well in terms of software complexity, however Bat algorithm had lesser lines of code compare to Flower Pollination algorithm.

## Conclusion

In this paper, Bat, Bee, Firefly and Flower Pollination algorithms were adapted and implemented for solving Job Shop Problem. The experimental result obtained on standard JSP showed that Bat Algorithm (BA) provides better results than Bee, Firefly, Flower Pollination Algorithms in most of the instances. Future works can be tailored towards hybridization of two algorithms to solve Job Shop Problem and observe whether there is any improvement to the results presented to this work.

## References

- [1] Abdel-Raouf O, Abdel-Baset M, El-Henawy I (2014a) An improved flower pollination algorithm with chaos. *Int J Educ Managt Eng* 4(2):1–8.
- [2] Binato, Silvio & J. Hery, W & M. Loewenstern, D. (2000). A Grasp for Job Shop Scheduling. *Essays and Surveys on Metaheuristics*. 15. 10.1007/978-1-4615-1507-4\_3.
- [3] BJ, Glover. (2008). Understanding Flowers and Flowering: An integrated approach. *Understanding Flowers and Flowering: An integrated approach*. 1-256.10.1093/acprof:oso/9780198565970.001.0001.
- [4] Bruker, P., Schlie, R. (1990): Job Shop Scheduling with Multi-Purpose Machine, *Computing*, Vol. 45, 1990.
- [5] Chaudhry, Imran Ali. (2012). Job Shop Scheduling Problem with Alternative Machines Using Genetic Algorithms. *Journal of Central South University of Technology*. 19.10.1007/s11771-012-1145-8.
- [6] 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.
- [7] Dubey HM, Pandit M, Panigrahi BK (2015b) A biologically inspired modified flower pollination algorithm for solving economic dispatch problems in modern power systems. *Cognit Comput* 7(5):594–608.
- [8] El-Bouri A., Azizi N., and Zolfaghari, S. (2007). A comparative study of a new heuristic based on adaptive memory programming and simulated annealing: The case of job shopscheduling. *European Journal of Operational Research*, 177(3), 1894-1910.
- [9] Fogel D.B., Fogel L.J. (1996) An introduction to evolutionary programming. In: Alliot JM., Lutton E., Ronald E., Schoenauer M., Snyers D. (eds) *Artificial Evolution*. AE 1995. Lecture Notes in Computer Science, vol 1063. Springer, Berlin, Heidelberg
- [10] Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.
- [11] Ge HW, Sun L, Liang YC, Qian F (2008). An Effective PSO and AISBased Hybrid IntelligentAlgorithm for Job-Shop Scheduling. *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*.38: 358-368.
- [12] H. P. Schwefel. (1981). "Numerical Optimization of Computer Models," John Wiley & Sons, Chichester. Volume 3, Issue 1
- [13] Hayes-Roth, Frederick. (1975). Review of "Adaptation in Natural and Artificial Systems by John H. Holland", The U. of Michigan Press, 1975. *Intelligence/sigart Bulletin - SIGART*. 1515. 10.1145/1216504.1216510.
- [14] J. Christopher Beck and Nic Wilson (2007): Proactive Algorithms for Job Shop Scheduling with Probabilistic Durations. *Journal of Artificial Intelligence Research* 28 (2007) 183–232.
- [15] John McCall (2004): Genetic algorithms for modelling and optimization. *Journal of Computational and Applied Mathematics* 184 (2005) 205 – 222.
- [16] Koza, J. R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: The MIT Press.
- [17] Lars Chittka, James D.Thomson, Nickolas M. Waser (1999): Flower Constancy, Insect Psychology, and Plant Evolution.
- [18] Liangshan Shao, Yuan Bai and Yunfei Qiu, Zhanwei Du (2012): Particle swarm optimization algorithm based on semantic relations and its engineering applications.
- [19] Lukasic S, Zak S (2009) Firefly algorithm for continuous constrained optimization task, ICCCI2009. In: Nguyen NT, Kowalczyk R, Chen SM (eds) *Lecture notes in artificial intelligence*, vol 5796. Springer, Berlin, pp 97–100.
- [20] Marta Castilho Gomes, Ana Paula Barbosa-Povoa and Augusto Q Novais (2005): Optimal scheduling for flexible job shop operation.
- [21] Mohamed Abdel-Basset and Laila Shawky (2008): Flower pollination algorithm: a comprehensive review.

- [22] Nishant Pathak, Sudhanshu Prakash Tiwari (2012): Travelling Salesman Problem Using Bee Colony With SPV. *International Journal of Soft Computing and Engineering (IJSCE)*
- [23] Pavlyukevich. I, (2007): Levy flights, non-local search and simulated annealing, *Journal of Computational Physics*, vol 226, no2, pp 1830-1844.
- [24] Rechenberg I. (1965) Cybernetic solution path of an experimental problem. Royal Aircraft Establishment, Farnborough p. Library Translation 1122.
- [25] Sana Jawarneh and Salwani Abdullah (2015): Sequential Insertion Heuristic With Adaptive Bee Colony Optimisation Algorithm for Vehicle Routing Problem With Time Windows.
- [26] Sato T and Hagiwara M (1997): Bee system Finding solution by a concentrated search. Proc. IEEE International Conference on Systems, Man and Cybernetics, 12-15, Hyatto Rlandoo, Rlandof, Lordia, USA. 3954-3959.
- [27] Teodorovic, D. & Dell'Orco, M. (2005): Bee colony optimization: A cooperative learning approach to complex transportation problems. *Advanced OR and AI Methods in Transportation*, pp. 51-60.
- [28] Xueni Qiu, Henry Y.K. Lau (2012): An AIS-based Hybrid Algorithm with PSO for Job Shop Scheduling Problem.
- [29] Yang X-S (2008) Nature-inspired metaheuristic algorithm. Luniver Press, Beckington.
- [30] Yang X-S (2009) Firefly algorithms for multimodal optimization, In: *Stochastic algorithms: foundations and applications*, SAGA, Lecture Notes in Computer Sciences, 5792, 169 178.
- [31] Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. *Int J BioInspired Comput* 2(2):78-84.
- [32] 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,
- [32] Zaid Abdi Alkareem Alyasseri, Mohammed Azmi Al-Betar, Ahamad Tajudin Khader, Mohammed A. Awadallah (2018): Variants of the Flower Pollination Algorithm: A Review.