

2024

Job Scheduling By Using Swarm Intelligent Algorithms: survey

Ruqaya Ahmed

College of Computer science & information technology, University of AL-Qadisiyah, Al-Qadisiyah, Iraq,
rugaya.ahmed@uomus.edu.iq

Luma S.Hasan

College of Computer science & information technology, University of AL-Qadisiyah, Al-Qadisiyah, Iraq,
luma.hasan@qu.edu.iq

Follow this and additional works at: <https://qjps.researchcommons.org/home>



Part of the [Biology Commons](#), [Chemistry Commons](#), [Computer Sciences Commons](#), [Environmental Sciences Commons](#), [Geology Commons](#), [Mathematics Commons](#), and the [Nanotechnology Commons](#)

Recommended Citation

Ahmed, Ruqaya and S.Hasan, Luma (2024) "Job Scheduling By Using Swarm Intelligent Algorithms: survey," *Al-Qadisiyah Journal of Pure Science*: Vol. 29 : No. 1 , Article 2.

Available at: <https://doi.org/10.29350/2411-3514.1224>

This Review is brought to you for free and open access by Al-Qadisiyah Journal of Pure Science. It has been accepted for inclusion in Al-Qadisiyah Journal of Pure Science by an authorized editor of Al-Qadisiyah Journal of Pure Science.

REVIEW

Job Scheduling By Using Swarm Intelligent Algorithms: Survey

Ruqaya Ahmed ^{a,b,*}, Luma S. Hasan ^a

^a College of Computer Science & Information Technology, University of AL-Qadisiyah, Al-Qadisiyah, Iraq

^b Al-Mustaqbal University College, 51001, Hillah, Babil, Iraq

Abstract

The job scheduling problem is used in industry, manufacturing, job planning, and the network environment to manage users jobs on the right machines with different limitations. There are three types of job scheduling problems. Firstly, job shop scheduling means each job is executed on a determined sequence of machines specified previously. While in the flow shop scheduling problem, each job can be carried on one idle sequencing using a job queue. In the third type, open shop, the sequence of jobs can be carried out on any free machine. This paper describes the various types of job scheduling problems and the various swarm intelligence algorithms, particularly cuckoo search algorithm, that can be used to solve them. In the future, build software to solve the job-shop problem by using cuckoo search optimization.

Keywords: Job scheduling problem, Swarm intelligence algorithms, Cuckoo search algorithm

1. Introduction

Scheduling is a way of controlling numerous queues of activities in order to reduce delays and improve system performance [24]. Job Scheduling focuses on users who have a set of jobs that must be completed sequentially on a large number of machines with a variety of constraints [1] (see Figs. 1 and 2).

Scheduling research has recently gotten much more attention in the business world, especially in its most typical manifestation, job-shop scheduling. Even though academics have made a lot of progress, there are still worries about transferring technology to meet the need for flexibility in modern manufacturing facilities. JSSP has attracted a lot of attention in the literature because the solution is made more complicated by the need to meet the opposing demands of batch and continuous production [2].

Most research in the recent decade has focused on novel problem-solving techniques such as

optimization algorithms. These techniques have shown to be a significant advancement in solving JSSP with less computing effort and more solid outcomes [2].

Firstly, job scheduling is designed using a mathematical model as well as operational research to find the best global solution. The researcher used artificial intelligence to solve the job scheduling problem in 1980, following the revolution in the development of intelligence algorithms with computer technology [24].

2. Problem definition

In the JSSP, a set "J" of "n" jobs (J₁, J₂, J₃, ... J_n) must be processed on a set "M" of "m" different machines. M₁, M₂, M₃, ... M_m. Job J_j consists of a sequence of M_J operations (O_{j1}, O_{j2}, O_{j3}, ... and O_{jmj}), which have to be scheduled in this order.

Furthermore, each operation can only be carried out by one of the m machines available. Operation "O_{jk}" has a processing time of "P_{jk}." The main

Received 8 March 2023; accepted 11 April 2024.
Available online 28 September 2024

* Corresponding author at: College of Computer Science & Information Technology, University of AL-Qadisiyah, Al-Qadisiyah, Iraq.
E-mail addresses: ruqaya.ahmed@uomus.edu.iq, com21.post5@qu.edu.iq (R. Ahmed), luma.hasan@qu.edu.iq (L.S. Hasan).

<https://doi.org/10.29350/2411-3514.1224>

2411-3514/© 2024 College of Science University of Al-Qadisiyah. This is an open access article under the CC-BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Algorithm 1 Original CS algorithm

```

Input: Population of nests  $x_i = (x_{i1}, \dots, x_{iD})^T$  for  $i=1 \dots Np, MAX\_FE$ .
Output: The best solution  $x_{best}$  and its corresponding value  $f_{min} = \min(f(x))$ .
1: generate initial host nest locations();
2:  $eval = 0$ ;
3: while termination condition not meet do
4:   for  $i = 1$  to  $Np$  do
5:      $x_j = generate\_new\_solution(x_i)$ ;
6:      $f_j = evaluate\_the\_new\_solution(x_j)$ ;
7:      $eval = eval + 1$ ;
8:      $j = \lfloor rand(0, 1) * Np + 1 \rfloor$ ;
9:     if  $f_j < f_j$  then
10:       $x_j = x_i; f_j = f_i$ ; // replace  $j$ -th solution
11:     end if
12:     if  $rand(0, 1) < p_a$  then
13:        $init\_nest(x_{worst})$ ;
14:     end if
15:     if  $f_i < f_{min}$  then
16:       $x_{best} = x_i; f_{min} = f_i$ ; // save the local best sol.
17:     end if
18:   end for
19: end while
    
```

Fig. 1. Original CS algorithm.

goal is to find an operating sequence for each machine to minimize a specific function of job completion times and ensure that no two operations are ever processed on the same machine simultaneously [18].

3. Job shop scheduling type

Job shop scheduling, often known as The JSP is an optimization issue where many types of

manufacturing jobs are allocated to machines at various times in order to reduce Makespan. Since 1956, a lot of research has been done on schedule because it directly affects the efficiency and cost of a manufacturing system's production [25].

JSP, on the other hand, is often an NP-complete combinatorial optimization issue. When scaling up a problem, traditional optimization approaches based on centralized or semi-distributed scheduling face major computational stability and time constraints [25].

Depending on the constraints and objectives of job scheduling, the problem can be classified as follows:

A. static job shop

A limited number of jobs, each with a preset order of priority for operations. [12], It must be processed on a limited number of machines. All jobs are released, and all machines are available at the start of the static JSSP. Each machine can only do one operation at a time, and no more than one machine can work on the same job at the same time [12].

B. Flexible job shop

Flexible job shop scheduling is the process of scheduling the operations in a flexible job shop, which is a type of manufacturing system where a wide variety of products are being produced and the processing steps for each product are different. In this type of system, there is often a large number of machines and a lot of flexibility in terms of which machine can be used to process a particular operation [11].

4. Metaheuristic optimisation algorithm

Metaheuristic algorithms are a kind of stochastic optimization method that does not use the surface gradient to optimize. To maximize fitness, these algorithms take inspiration from a number of sources, while the majority of them are nature-inspired [21]. Due their various applications in engineering and artificial intelligence, they have been extensively researched.

Swarm intelligence (SI) is a field of Metaheuristic algorithms that refers to the collective behavior shown by social insect swarms and has attracted the interest of scientists. Without a centralized supervisor, swarms can solve problems that are too hard for individual insects to handle. Then, researchers devised methods using this concept to solve combinatorial optimization issues. These include proven techniques such as Genetic Algorithm (GA),

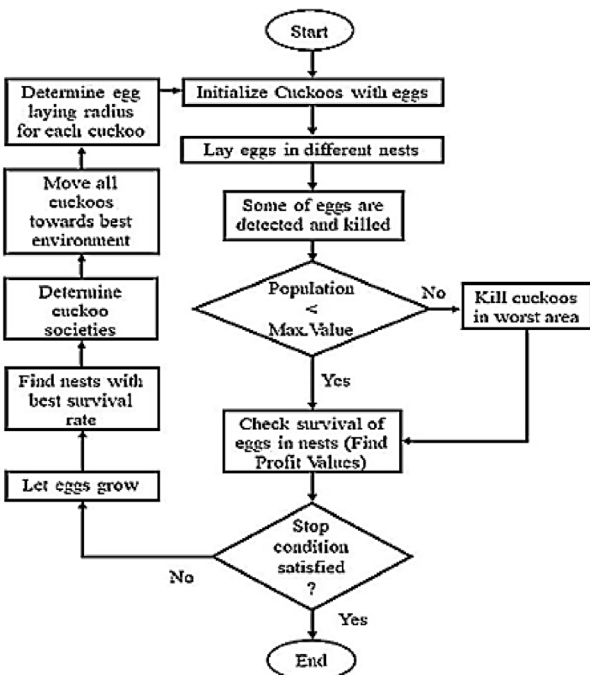


Fig. 2. Flowchart of Cuckoo Search Algorithm [2].

Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Cuckoo Search Optimization (CSO), among others, which are briefly described below:

A. Genetic algorithm (GA)

A genetic algorithm (GA) is a search algorithm that uses principles of natural evolution, such as reproduction, mutation, and selection, to find approximate solutions to optimization problems. GAs are useful for problems where it is difficult or impossible to find an exact solution, and can be applied to a wide range of fields, including machine learning, engineering, and finance [21]. GAs work by generating a population of candidate solutions and iteratively improving them through a process of selection, crossover, and mutation until a satisfactory solution is found.

B. Ant colony algorithm (AC)

Ant Colony Optimization algorithm is a novel simulated a evolutionary heuristic optimization method that has been successfully used to solve several NP-hard combinatorial optimization problems. In 1996, Italian academic M. Dorigo devised the ACO algorithm based on ant food-seeking behavior. The ant is a basic computational agent that creates a solution to the issue iteratively [5]. The ants release pheromones as they advance; when they reach a crossing they have never traversed before, they will select a route at random and emit pheromones according to the path length. By detecting the pheromone on the way, Follower ants follow the other ants' tracks to the food source. As the process continues, most ants are more inclined to take the shortest road with the most pheromones; on this occasion, a positive feedback loop arises; this mechanism guarantees that sound information is kept and ants may finally choose an ideal path. In this situation, there are an increasing number of pheromones along this route. The quantity of information on pathways steadily diminishes over time, until the road most ants choose becomes the ideal path. When a combinatorial issue can be defined, an ACO algorithm may be applied to it [9].

C. Bat optimization algorithm (BA)

The bat optimization algorithm (BOA) is a Meta-heuristic optimization algorithm inspired by the echolocation behavior of bats. It is a population-based algorithm used to find a function's global minimum or maximum [16]. The algorithm works by simulating the behavior of bats, where each bat is

represented by a set of decision variables corresponding to its position in the search space. The bats emit vocalizations and adjust their frequency and loudness based on the quality of the solution at their current position. The bats also use echolocation to locate food sources, which in the context of the algorithm, corresponds to the optimal solution. The process is repeated until a satisfactory resolution or predetermined stopping criteria is met. The BOA is a simple and effective algorithm successfully applied to various optimization problems [16].

D. Firefly algorithm (FA)

The firefly algorithm (FA) is a metaheuristic optimization algorithm inspired by fireflies' flashing behavior. It is a population-based algorithm used to find a function's global minimum or maximum. The algorithm works by simulating the behavior of fireflies. Each firefly is drawn to other fireflies that are brighter, and the brightness of a firefly depends on the value of the objective function at its current position [19]. The fireflies move around in the search space, and their positions are updated based on their attraction to other fireflies and the influence of random noise. The process is repeated until a satisfactory solution, or a predetermined stopping criterion is met. The FA is a simple and effective algorithm that has been successfully applied to a variety of optimization problems.

5. Cuckoo search algorithm

Cuckoo search is a meta-heuristic approach inspired by the cuckoo bird, a "brood parasite" bird. It never builds its own nest, preferring to lay eggs in the nest of another host bird. The cuckoo is the most well-known brood parasite. Some host birds may engage the invading cuckoo directly. If the host bird detects that the eggs are not its own, it will either remove them from the nest or leave the nest and build a new one. Each egg in a nest represents a solution, and the cuckoo egg represents a unique and appropriate solution. The acquired answer is a new one based on the previous one with modified features. Each nest has one cuckoo egg in its most basic form, so Multiple eggs, each representing a different variety of solutions, will be placed in each nest. In structural engineering, Scheduling issues are effectively addressed by the use of CS. as well as design optimization problems. Speech reorganization, task scheduling, and global optimization are only a few examples. Cuckoo search idealizes similar breeding behavior and may be used to address a wide range of optimization issues [7].

The most important properties of CS are described below:

5.1. Random walk

A random walk is a series of continuous, random steps. Assuming that (SN) is the sum of the sequentially random steps (X_i), we can assume that (SN) is also generated randomly, as shown in the equation. Here are the most essential parts of CS:

$$S_N = \sum_{i=1}^N X_i = X_1 + \dots + X_N$$

X_i , on the other hand, is a random step obtained from a "random distribution." The following nested equation may be used to express the connection:

$$S_N = \sum_{i=1}^{N-1} X_i + X_N = S_{N-1} + X_N$$

This signifies that the next state (SN) is solely dependent on the following:

- a. The Present Situation (SN-1)
- b. The movement or transitions (XN) between the current and following states.

The size or length of the step in the random walk might be constant or variable [2].

5.2. Lévy flights mechanism

Animals in the wild seek food at random or near-random intervals. The next step in an animal's foraging path is defined by both the animal's current location/state and the transition probability to the next site. The chosen pathways are quantitatively represented using probability. Several studies have shown that the flying behavior of several animals and insects is similar to Lévy flights. A Lévy flight is a random walk with step lengths specified by a probability distribution with a heavy tail. The distance from the random walk's origin tends to have a stable distribution after a large number of steps [2].

$$X_i^{(t+1)} = X_i(t) * \alpha \oplus \text{levy}(\lambda)$$

The step size is represented by α where $\alpha > 0$. Most of the time, assume it equals 1. When we talk about product \oplus , we're talking about entry-wise multiplication, also called an exclusive OR operation. A random walk with steps selected from a levy distribution is called a levy flight.

$$\text{levy} \sim u = t^{-\lambda} \quad (1 < \lambda \leq 3)$$

when both the standard deviation and the mean are infinitely large. Cuckoo behavior may be viewed as

a random walk process with a power-law distribution of step length and a heavy tail [4].

6. Cuckoo search applications

CS has been used successfully in various fields of optimization and computational intelligence. Cuckoo search, for example, works better than other algorithms for certain recurrent optimization tasks in engineering design, such as designing a spring or a welded beam [23].

Also, it has been shown that a modified cuckoo search is a very effective way to solve nonlinear problems like mesh generation. The cuckoo search was used to train spiking neural network models. The cuckoo search was used to optimize semantic web service composition processes. Furthermore, they used cuckoo search to find the best design for a reliable embedded system and to design steel frames that worked well. In milling operations, CS was used to choose the best machine parameters, which led to better results and gave the Gaussian process an alternative to the cuckoo search.

On the other hand, a discrete cuckoo search algorithm has been developed to handle nurse scheduling problems. Cuckoo led to the creation of separate pathways for software testing and data generation. Cuckoo search is particularly effective in data fusion and wireless sensor networks. Also, an efficient way to solve Knapsack problems has been found by combining a cuckoo search variation with a quantum-based technique.

7. Researcher studies

Ouaarab et al. [22] focuses on combinatorial optimization problems. The authors proposed a discrete cuckoo search to explore space, they go on a global random walk. In fact, the suggested solution gives a space solution as well as how to move from one solution to another via Levy flights. The proposed maximum optimal solution (1784) was achieved.

Singh et al. [18] provided a solution to the job shop scheduling problem. The authors suggested a unique hybrid algorithm that incorporated the cuckoo search optimization approach with an assorted individual enhancement scheme. When the algorithm converges, the best, or, say, fittest, host is the final solution to the optimization problem being solved. The recommended optimum maximum solution was achieved (1,292).

Kamoona et al. [8] focused on the weak convergence speed of the cuckoo search (CS) algorithm and the need for a good balance of local and global search. Depending on Gaussian diffusion random walks and a greedy selection approach, the authors

suggested an enhanced cuckoo search (ECS) algorithm. In contrast, The greedy selection approach has been removed from the enhanced cuckoo search algorithm resolves the CS algorithm's exploitation power in obtaining the optimal solution under the effect of Gaussian diffusion random walks. The proposed solution was successful ($-3.86E+00$).

Pandan et al. [11] The author suggested a cuckoo search optimization (CSO) algorithm based on simulation. The simulation-based CSO technique was successfully used to solve problems with flexible job shop scheduling. The algorithm was put through its paces on a small scenario involving transportation time and production quantity. Makespan is a performance metric. The suggested solution worked (8). Quantity of production.

Tagtekin et al. [20] A genetic algorithm (GA) was suggested by the authors. When resolving JSSP, use a heuristic technique called Genetic Algorithm (GA) to find the most robust solution for various scheduling and maximum utilisation problems. The variety of a population will almost certainly increase, and the time taken by the crossover function will decrease. When implemented in the CI system, the suggested solution (20%) was obtained.

Shareh et al. [16] Based on ColReuse, the Bat Algorithm (BA) was created to solve job scheduling problems in open shops. The proposed strategy makes use of evolutionary processes, which may lead to an acceptable solution to the problem in a reasonable amount of time. The suggested answer was achieved when applied to the 50x5-4 benchmark dataset (7217).

Da Silva et al. [5] To solve JSSP, the authors suggested Ant Colony Optimization (ACO), which provides the optimal Makespan. According to the suggested approach, the algorithm can achieve a global optimum in both ft06 and la01 situations. The suggested solution was implemented. In ft06, the average Makespace was $[56.57]_{(-0.5)(+0.49)}$; in la01, the average Makespace was $[673.07]_{(-2.08)(+3.99)}$.

Singh et al. [17] The authors focused on solving job scheduling using a hybrid algorithm that used genetic and cuckoo search algorithms. A hybrid algorithm starts with a set of jobs and machines. A hybrid algorithm is used to schedule tasks. The suggested method indicates that the hybrid algorithm outperforms the genetic algorithm. When applied to 3 machines and 6 jobs, the suggested solution achieved (5.3191) time taken by the genetic algorithm and (0.5586) time taken by the hybrid algorithm.

Bruin et al. [3]. Focused on solving submit jobs to grid computing environments. The authors proposed

my_condor_submit (MCS) for solving accessing grid resources; Starting with parameter sweep information, producing multiple sets of input files, submitting and executing all jobs, returning data, and finally collecting particulars of interest data and creating a single graph displaying results from all jobs, a full process may be created. The plan is to give users a group of machines to which they can send their MCS jobs. The proposed solution was implemented (XML files with pressure and volume values).

Rabiee et al. [13] In grid computational design, the authors proposed the Cuckoo Optimization Algorithm (COA). The authors employed COA on JSP to analyze the performance of the JSS algorithm in order to solve job scheduling. The suggested solution includes an algorithm that attempts to find the most suitable solution. The suggested solution was performed in (6) seconds.

Ashok et al. [6]. The authors proposed a gradient-based cuckoo search (GBCS) for solving an optimisation problem; the gradient was It is easily accessible and may be used to enhance the numerical performance of stochastic optimization techniques, particularly the quality and accuracy of the optimal global solution. The alteration he suggested was quietly implemented in the algorithm by modifying the direction of the local random walk of cuckoos. The suggested solution was implemented ($7.2758E-13$) Mean.

Qinghua et al. [10]. The authors proposed a genetic algorithm for solving JSP, To increase the efficiency of big data analytics, we present a genetic algorithm-based job scheduling model for big data analytics applications. The recommended approach achieved a time of 740.56 s at a cost of 0.56.

Saxena et al. [14] The authors presented an Ant Colony Optimization algorithm (ACO) for solving JSSP; a task scheduler dynamically modifies its scheduling approach based on the changing environment and job kinds. The suggested solution was implemented (2177.9954).

8. Scheduling by cuckoo search

The template is created for, but not limited to, the following six Job scheduling problems, a combinatorial optimisation problem which involves assigning tasks to the machines so that all tasks should get resources on time and complete their execution without affecting other tasks. babukartik and Dhavachelvan (2012) [1] developed a hybrid algorithm consisting of ACO and CS algorithms for scheduling jobs. Using Lévy flights, the CS algorithm was used to increase the local search in solution space. The schedule was run using a scheduler, and it found

that the Makespan was lesser when using a hybrid algorithm than in ACO alone. Combining PSO and CS, two nature-inspired Metaheuristic algorithms, Selvi (2019) [15] proposed an improved cuckoo search method for resolving job scheduling problems. The advantage of creating a velocity value by the PSO algorithm was used for generating new solutions, increasing the efficiency of the improved approach in job scheduling and achieving the minimum Makespan time. Computational results demonstrate that the ICS approach is better than the other simplex evolutionary algorithm. Satyendra designed a hybrid algorithm that combines the greatest features of both the genetic and cuckoo search algorithms. For job scheduling problems aimed at minimizing Makespan. The disadvantage of GA being trapped in local optimal was overcome using the CS algorithm along with GA. The result of analysis showed the performance of the suggested algorithm was better than that of the genetic algorithm.

9. Conclusion

Up until now, most research has been focused on improving the performance and updating the artificial algorithms to find the best way to solve the job shop scheduling problem.

All the researchers applied swarm intelligence or combined two methods to summarize the advantages and disadvantages of the methods. This paper presented the job scheduling types and many of the swarm intelligence algorithms that were used to solve them, especially the cuckoo search algorithm.

Acknowledgements

I extend my sincere thanks to Dr. Lama Salal Hasan, and I also extend my sincere thanks to the Al-Mustaqbal University College for its financial support in this research.

References

- [1] Adibi MA, Zandieh M, Amiri M. Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Syst Appl* 2010;37(1):282–7.
- [2] Al-Abaji MA. Cuckoo search algorithm: review and its application. *Tikrit J Pure Sci* 2021;26(2):137–44.
- [3] Bruin RP, White TOH, Walker AM, Austen KF, Dove MT, Tyler RP, et al. Job submission to grid computing environments. *Concurr Comput Pract Ex* 2008;20(11):1329–40.
- [4] Cao ZC, Lin CR, Zhou MC. A knowledge-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility. *IEEE Trans Autom Sci Eng* 2019;18(1):56–69.
- [5] da Silva AR. Solving the job shop scheduling problem with ant colony optimization. *arXiv preprint arXiv:2209.05284*; 2022.
- [6] Fateen S-EK, Bonilla-Petriciolet A. Gradient-based cuckoo search for global optimization. *Math Probl Eng* 2014;2014.
- [7] Joshi AS, Kulkarni O, Kakandikar GM, Nandedkar VM. Cuckoo search optimization-a review. *Mater Today Proc* 2017;4(8):7262–9.
- [8] Kamoona AM, Patra JC, Stojcevski A. An enhanced cuckoo search algorithm for solving optimization problems. In: 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE; 2018.
- [9] Liu C-Y, Zou C-M, Wu P. A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing. In: 2014 13th international symposium on distributed computing and applications to business, engineering and science. IEEE; 2014.
- [10] Lu Q, Li S, Zhang W, Zhang L. A genetic algorithm-based job scheduling model for big data analytics. *EURASIP J Wirel Commun Netw* 2016;2016(1):1–9.
- [11] Phanden RK, Saharan L, Erkoyuncu JA. Simulation based cuckoo search optimization algorithm for flexible job shop scheduling problem. In: Proceedings of the international conference on intelligent science and technology; 2018.
- [12] Qiu X, Lau HY. An AIS-based hybrid algorithm for static job shop scheduling problem. *J Intell Manuf* 2014;25(3):489–503.
- [13] Rabiee M, Sajedi H. Job scheduling in grid computing with cuckoo optimization algorithm. *Int J Comp Appl* 2013;62(16).
- [14] Saxena D, Chauhan RK, Kait R. Dynamic fair priority optimization task scheduling algorithm in cloud computing: concepts and implementations. *Int J Comput Netw Inf Secur* 2016;8(2):41.
- [15] Selvi C, Sivasankar E. A novel optimization algorithm for recommender system using modified fuzzy c-means clustering approach. *Soft Comput* 2019;23(6):1901–16.
- [16] Shareh MB, Bargh SH, Hosseinabadi AA, Slowik A. An improved bat optimization algorithm to solve the tasks scheduling problem in open shop. *Neural Comput Appl* 2021;33(5):1559–73.
- [17] Singh S, Kurmi J, Tiwari SP. A hybrid genetic and cuckoo search algorithm for job scheduling. *Int J Sci Res Pub* 2015; 5(6):1–4.
- [18] Singh S, Singh KP. Cuckoo search optimization for job shop scheduling problem. In: Proceedings of fourth international conference on soft computing for problem solving. Springer; 2015.
- [19] Świtalski P, Bolesta A. Firefly algorithm applied to the job-shop scheduling problem. *Studia Info Syst Info Technol* 2021;25(1–2):87–100.
- [20] Tağtekin B, Öztürk MU, Sezer MK. A case study: using genetic algorithm for job scheduling problem. *arXiv preprint arXiv:2106.04854*; 2021.
- [21] Wong W, Ming CI. A review on metaheuristic algorithms: recent trends, benchmarking and applications. In: 2019 7th international conference on smart computing & communications (ICSCC). IEEE; 2019.
- [22] Yang X-S, Deb S. Cuckoo search via Lévy flights. In: 2009 World congress on nature & biologically inspired computing (NaBIC). Ieee; 2009.
- [23] Yang X-S, Deb S. Cuckoo search: recent advances and applications. *Neural Comput Appl* 2014;24(1):169–74.
- [24] Yu Y. A research review on job shop scheduling problem. In: E3S web of conferences. EDP Sciences; 2021.
- [25] Zhang J, Ding G, Zou Y, Qin S, Fu J. Review of job shop scheduling research and its new perspectives under Industry 4.0. *J Intell Manuf* 2019;30(4):1809–30.