Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Job Scheduling By Using Swarm Intelligent Algorithms: survey

Ruqaya Ahmed[a,b], Dr.Luma S.Hasan[a]

[a]*College of Computer science & information technology, University of AL-Qadisiyah, Al-Qadisiyah, Iraq*

Email: com21.post5@qu.edu.iq, luma.hasan@qu.edu.iq

[b]*Al-Mustaqbal University, 51001 Hillah, Babil, Iraq.*

Email: rugaya.ahmed@uomus.edu.iq

ARTICLE INFO

Article history:
Received: 25 /04/2023
Rrevised form: 05 /06/2023
Accepted : 14 /06/2023
Available online: 30 /06/2023

Keywords:

Job Scheduling Problem, Swarm Intelligence Algorithms , Cuckoo Search Algorithm, Genetic Algorithem, Firefly Algorithem, Ant Colony Algorithm, Bat Optimization Algorithm,

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. 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 used to solve them.

https://doi.org/10.29304/jqcm.2023.15.2.1238

1. Introduction

Scheduling is a way of controlling numerous queues of activities to reduce delays and improve system performance [23]. 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]. 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]. However, the researches 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 [23].

2. Aim and objectives

The JSSP involves a set "J" of "n" jobs (J_1, J_2, J_3,... Jn) that need to be processed on a set "M" of "m" distinct machines (M_1, M_2, M_3,... Mm). Each job J_j comprises a sequence of M_J operations (O_j1, O_j2, O_j3,... and O_jmj), which must be scheduled in their specified order. Additionally, each operation can be executed by only one of the available m machines, and has a processing time of "P_jk." The primary objective is to determine an operating sequence for each machine that minimizes a specific function of job completion times while ensuring that no two operations are processed on the same machine simultaneously [17].

3. Job Shop Scheduling Type

The JSP, or Job Shop Scheduling, is a problem of optimization that involves allocating various manufacturing jobs to machines at different times with the aim of minimizing Makespan. This issue has been extensively researched since 1956, as it has a direct impact on the efficiency and production cost of a manufacturing system. [14]. Combinatorial optimization problem of JSP, on the other hand, is frequently NP-complete. As the problem size increases, conventional optimization methods that rely on centralized or semi-distributed scheduling encounter significant computational and time limitations. [24]. In the same context, , the problem can be classified depending on the constraints and objectives of job scheduling in to :

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 Optimization 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 [20]. 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), 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. 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. In general, genetic algorithm (GA) is more suitable for optimization (maximizing and minimizing) problems in numerical and random sequences with much better solutions in a short time than many other mathematical modelling alternatives as shown in figure 1. The first historical appearance of decimal number system is explained in detail. The application of GA mutation and crossover operations depend on random variability, so different random selections are conveniently explained to GA procedures. The major components of GA procedure are optimization, error minimization, fitness, target function, initial population, mutation, and crossover procedures. Although there are random elements in the method itself, GA can reach the absolute optimization solution in the shortest time. The GA method is easy to understand by everyone as it can reach the result with only arithmetic calculations without requiring detailed and heavy mathematics. [20].

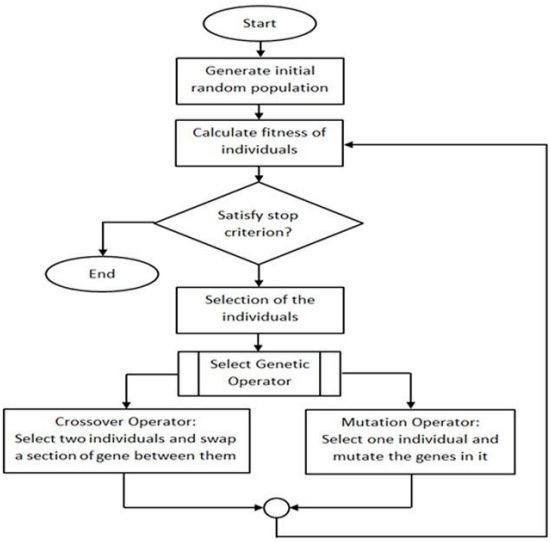

Fig. 1 - flowchart of GA Algorithm

b. Ant Colony Algorithm(AC)

Ant Colony Optimization algorithm is a novel simulated an 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 as shown in Figure 2. 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].

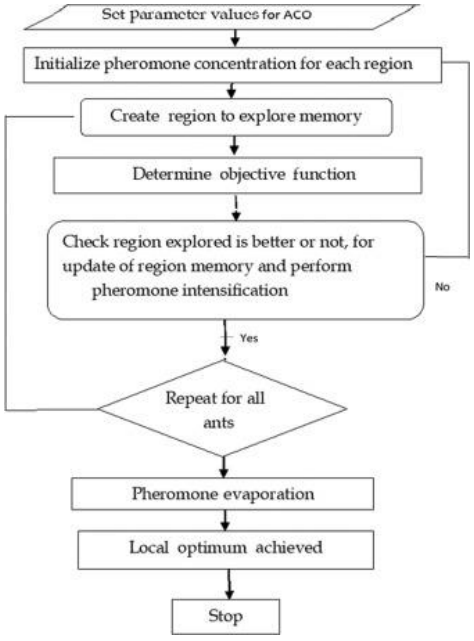

Fig. 2 - flowchart of ACO Algorithm

c. Bat Optimization Algorithm (BA)

The bat optimization algorithm (BOA) is a Metaheuristic 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 as shown in Figure 3. 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].

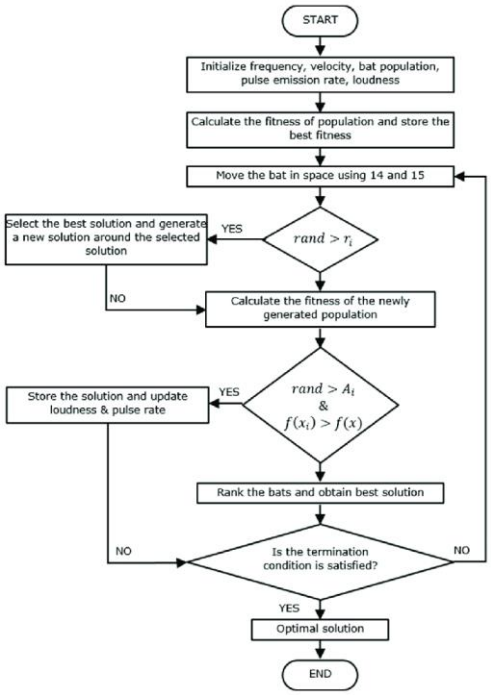

Fig. 3 - flowchart of BA Algorithm

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 behaviour of fireflies as shown in Figure 4. 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 [18]. 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 [18].

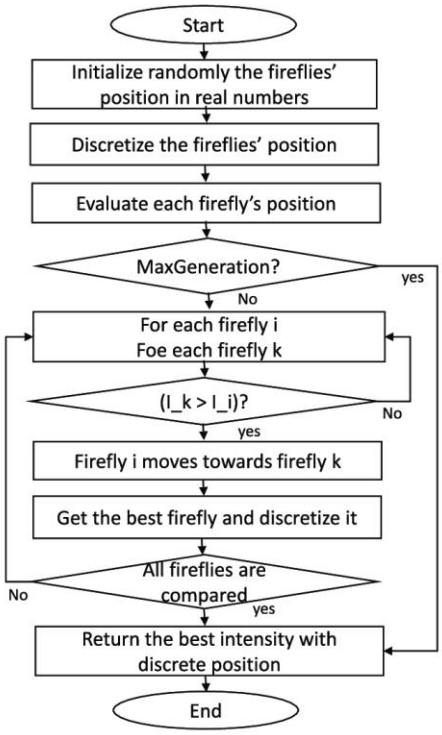

Fig. 4 - flowchart of FA Algorithm

4.1 Cuckoo Search Algorithm

Cuckoo search is a meta-heuristic optimization method that takes inspiration from the behavior of the cuckoo bird, which is a "brood parasite" bird that lays its eggs in the nests of other birds. Rather than building its own nest, the cuckoo relies on host birds to incubate and rear its young. In some cases, the host bird will detect the foreign eggs and either remove them or abandon the nest. The cuckoo search algorithm uses this behavior as a model for searching for solutions to optimization problems. Each cuckoo egg represents a unique and potentially optimal solution, and multiple eggs are placed in each nest to improve the chances of finding a suitable solution as shown in Figures 6 and 5. This approach has been successfully applied to a wide range of optimization problems, including scheduling, design, speech recognition, and global optimization, among others. The cuckoo search algorithm offers an efficient and effective means of solving complex optimization problems by mimicking the natural behavior of the cuckoo bird. [7]. The most important properties of CS are described below:

A- Random Walk

A random walk is a series of continuous, random steps. Assuming that (SN) is the sum of the sequentially random steps (Xi), can assume that (SN) is also generated randomly, as shown in the equation.

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

Here are the most essential parts of CS:
Xi, on the other hand, is a random step obtained from a "random distribution." The following nested equation may be used to express the connection:

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

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

1. The Present Situation (SN-1).
2. The movement or transitions (XN) between the current and following states.
3. The size or length of the step in the random walk might be constant or variable [2].

B- Lévy Flights Mechanism

In the wild, animals typically search for food randomly or semi-randomly. The subsequent step in their foraging path is determined by the current state/location of the animal and the probability of transitioning to the next site. The selected routes are quantitatively represented using probability. Numerous studies have indicated that the flight behavior of various animals and insects resembles that of

Lévy flights, which are random walks with step lengths determined by a probability distribution that possesses a heavy tail. After a large number of steps, the distance from the origin of the random walk tends to have a stable distribution [2].

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

The step size is denoted by α, where α > 0, with a default assumption of 1. The product symbol $\oplus$ refers to entry-wise multiplication, which is also known as an exclusive OR operation.
A random walk that employs steps drawn from a Lévy distribution is known as a Lévy flight:

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

In situations where both the standard deviation and the mean are infinitely large, the foraging behavior of cuckoos can be interpreted as a random walk process that utilizes a power-law distribution of step lengths and a heavy tail [4].

1: **begin**
2: Set iteration $\boldsymbol{t = 1}$.
3: Define problem dimension.
4: Initialize the population of nests.
5: **while** (termination condition not reached) **do**
6: Randomly generate cuckoo egg by Levy flight.
7: Evaluate the cuckoo fitness.
8: Randomly select a nest among available host nests.
9: The worst nests are abandoned and new ones are built.
10: Keep the best solutions or nests.
11: Rank the fitness and find the current best solution.
12: Set iteration $\boldsymbol{t = t + 1}$.
13: **end while**
14: **end**

Fig. 5 – Original CS algorithm

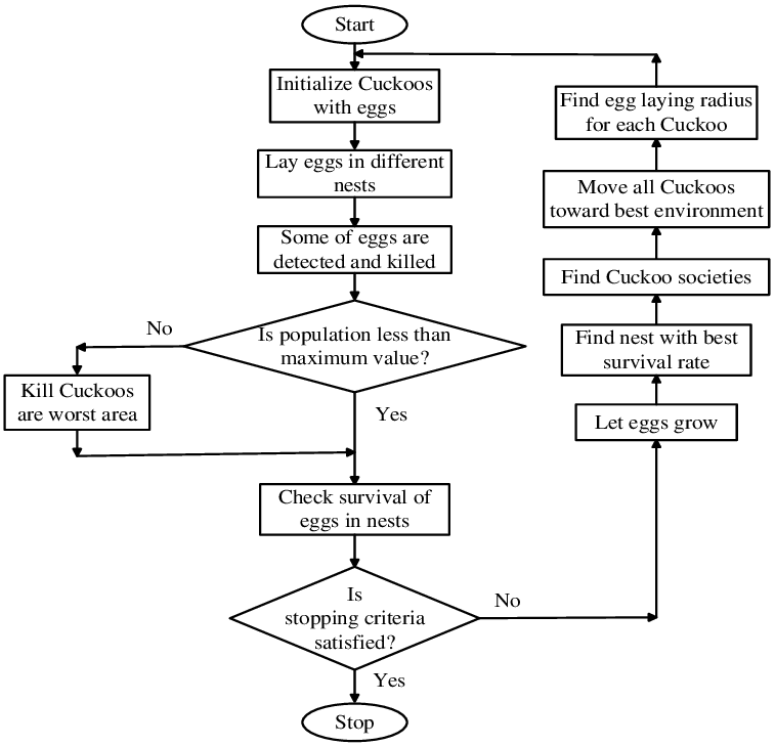

Fig. 6 - Flowchart of Cuckoo Search Algorithm.

4.1.1 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 [22]. 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.

5 Researcher Studies

Ouaarab et al. [21] 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. [17] The authors proposed a resolution for the job shop scheduling problem using a unique hybrid algorithm that combined the cuckoo search optimization approach with an assorted individual enhancement scheme. Upon convergence, the algorithm yielded the best or fittest host as the final solution to the optimization problem. The recommended optimal maximum solution was attained, reaching a value of 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. [19] 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. [7] 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, present a genetic algorithm-based job scheduling model for big data analytics applications. The recommended approach achieved a time of 740.56 seconds 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).

6 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.

7 Conclusions

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 for its financial support in this research.

References

[1] Adibi, M., et al. (2010). "Multi-objective scheduling of dynamic job shop using variable neighborhood search." Expert Systems with Applications 37(1): 282-287.

[2] Al-Abaji, M. A. (2021). "Cuckoo search algorithm: review and its application." Tikrit Journal of Pure Science 26(2): 137-144.

[3] Bruin, R. P., et al. (2008). "Job submission to grid computing environments." Concurrency and Computation: Practice and Experience 20(11): 1329-1340.

[4] Cao, Z., et al. (2019). "A knowledge-based cuckoo search algorithm to schedule a flexible job shop with sequencing flexibility." IEEE Transactions on Automation Science and Engineering 18(1): 56-69.

[5] da Silva, A. R. (2022). "Solving the Job Shop Scheduling Problem with Ant Colony Optimization." arXiv preprint arXiv:2209.05284.

[6] Fateen, S.-E. K. and A. Bonilla-Petriciolet (2014). "Gradient-based cuckoo search for global optimization." Mathematical Problems in Engineering 2014.

[7] Joshi, A. S., et al. (2017). "Cuckoo search optimization-a review." Materials Today: Proceedings 4(8): 7262-7269.

[8] Kamoona, A. M., et al. (2018). An enhanced cuckoo search algorithm for solving optimization problems. 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE.

[9] Liu, C.-Y., et al. (2014). A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing. 2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, IEEE.

[10] Lu, Q., et al. (2016). "A genetic algorithm-based job scheduling model for big data analytics." EURASIP journal on wireless communications and networking 2016(1): 1-9.

[11] Phanden, R. K., et al. (2018). Simulation based cuckoo search optimization algorithm for flexible job shop scheduling problem. Proceedings of the international conference on intelligent science and technology.

[12] Qiu, X. and H. Y. Lau (2014). "An AIS-based hybrid algorithm for static job shop scheduling problem." Journal of Intelligent Manufacturing 25(3): 489-503.

[13] Rabiee, M. and H. Sajedi (2013). "Job scheduling in grid computing with cuckoo optimization algorithm." International Journal of Computer Applications 62(16).

[14] Saxena, D., et al. (2016). "Dynamic fair priority optimization task scheduling algorithm in cloud computing: concepts and implementations." International Journal of Computer Network and Information Security 8(2): 41.

[15] Selvi, C. and E. Sivasankar (2019). "A novel optimization algorithm for recommender system using modified fuzzy c-means clustering approach." Soft Computing 23(6): 1901-1916.

[16] Shareh, M. B., et al. (2021). "An improved bat optimization algorithm to solve the tasks scheduling problem in open shop." Neural Computing and Applications 33(5): 1559-1573.

[17] Singh, S. and K. P. Singh (2015). Cuckoo search optimization for job shop scheduling problem. Proceedings of Fourth International Conference on Soft Computing for Problem Solving, Springer.

[18] Świtalski, P. and A. Bolesta (2021). "Firefly algorithm applied to the job-shop scheduling problem." Studia Informatica. System and information technology 25(1-2): 87-100.

[19] Tağtekin, B., et al. (2021). "A Case Study: Using Genetic Algorithm for Job Scheduling Problem." arXiv preprint arXiv:2106.04854.

[20] Wong, W. and C. I. Ming (2019). A review on metaheuristic algorithms: recent trends, benchmarking and applications. 2019 7th International Conference on Smart Computing & Communications (ICSCC), IEEE.

[21] Yang, X.-S. and S. Deb (2009). Cuckoo search via Lévy flights. 2009 World congress on nature & biologically inspired computing (NaBIC), Ieee.

[22] Yang, X.-S. and S. Deb (2014). "Cuckoo search: recent advances and applications." Neural Computing and Applications 24(1): 169-174.

[23] Yu, Y. (2021). A Research Review on Job Shop Scheduling Problem. E3S Web of Conferences, EDP Sciences.

[24] Zhang, J., et al. (2019). "Review of job shop scheduling research and its new perspectives under Industry 4.0." Journal of Intelligent Manufacturing 30(4): 1809-1830.