JOURNAL OF APPLIED INTELLIGENT SYSTEMS & INFORMATION SCIENCES

VOL. 5, ISSUE 1, PP. 23-32, FEBRUARY 2024.
Available at: https://journal.research.fanap.com/
DOI: https://doi.org/10.22034/JAISIS.2024.423324.1075

ENERGY-CONSCIOUS FLEXIBLE JOB SHOP SCHEDULING USING METAHEURISTIC ALGORITHMS

Atefeh Bagheri Verkiani [1], Parsa Fallah Sheikhlari [2], Seyed Habib A.Rahmati [3,*]

[1,3] Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran,
[2] Jawaharlal Nehru Technological University Hyderabad, India,
[2] Firooz Innovation and Research Center, Firooz Hygienic Group, Qazvin, Iran.

ABSTRACT

Within the landscape of manufacturing optimization, this research grapples with the intricate Flexible Job Shop Scheduling Problem (FJSP), particularly focusing on energy-conscious practices. In the production management context, FJSP specifies the task allocation to machines and determines the relevant task sequences while the energy-saving perspective is to handle the green-oriented concerns. Such a setting will result in an NP-hard problem. To this end, the study employs Genetic Algorithm (GA) and Simulated Annealing (SA) as metaheuristic tools to address the FJSP's challenges, emphasizing sustainability in industrial scheduling. Traditional models have often overlooked energy considerations, but in response to the growing need for environmentally friendly practices, this research explores avenues for achieving near-optimal solutions in the complex industrial scheduling domain. It contributes to the advancement of scheduling techniques in complex industrial settings. To capture the underlying uncertainty of the given domain, the energy consumption of machines is computed under a fuzzy modeling formulation.

KEYWORDS: Flexible Job Shop Problem (FJSP), Sequencing, Fuzzy Problems, Meta-heuristic Algorithms.

1. INTRODUCTION

Today, production flexibility has emerged as a competitive advantage, enabling systems to respond quickly to unpredictable changes. Scheduling problems find applications in various scientific fields, ranging from economics and computer engineering to diverse production environments. One of the well-known and fundamental issues in production environments is the Flexible Job Shop Scheduling Problem (FJSP). The problem of workshop scheduling, in its classical form, involves the scheduling of a set of tasks on a group of machines, aiming to optimize multiple performance indicators while considering the constraint that each task follows a fixed and predetermined path and is processed on all machines. The FJSP problem represents one of the flexible production policies, owing to the adaptability of the machines in this system and their ability to produce different products and handle various orders. This study is based on various theories, including flexibility theory, production efficiency, and more. Machine scheduling is one of the essential and well-recognized topics that has been extensively researched. Companies are required to produce diverse products tailored to specific

[*] *Corresponding Author, Email:* sd_rahmati@qiau.ac.ir

RECEIVED: 01 NOVEMBER 2023; ACCEPTED: 01 JANUARY 2024; PUBLISHED ONLINE: 28 FEBRUARY 2024

customer specifications and deliver them on time. In the current business environment, the competition among manufacturing companies is determined by their ability to promptly respond to rapid changes in the market and produce high-quality products with lower costs. In today's world, operational factors alone are no longer sufficient in production management, and constraints or values related to sustainable development must also be considered. Therefore, considering specialized energy-related factors can contribute to the rational development of these issues. Job Shop Scheduling Problem (JSP) is one of the most significant and widely used manufacturing scheduling policies, which aims to schedule a set of jobs on a given set of machines, each with its processing and setup times. JSP is considered one of the most well-known and extensively studied production policies.

In the FJSP scenario, unlike the JSP, jobs can be processed on a set of proposed machines capable of performing their respective operations. This flexibility in the real-world application of FJSP compared to JSP significantly increases its practicality and constitutes a major difference or extension of the FJSP problem compared to the JSP. Regarding the structure of the FJSP problem, it can be noted that this problem is composed of two subproblems. The first subproblem deals with the assignment of various job operations to machines. This assignment is done in such a way that, for each operation, the corresponding allocated machine is selected from a set of machines capable of performing that specific operation. The second sub-problem, following the first sub-problem, focuses on determining the sequence of performing different operations assigned to each machine. The existence of these sub-problems, especially the first sub-problem and the dependency that the second sub-problem has on the first sub-problem, has increased the complexity of the FJSP compared to the JSP. However, it has also significantly expanded its practical applications in real-world scenarios. In the rest of the paper, the relevant studies are reviewed in Section 2. Model formulation and solution methodology are described in Sections 3 and 4, respectively. Then, Section 5 discusses the results while Section 6 is devoted to the conclusion.

2. LITERATURE REVIEW

This section is to review related studies in the context of FJSP and green production as the two key concepts of the present paper. In terms of FJSP, since its introduction, scholars have tried to make problem setting closer to real-world cases while enhancing the quality of obtained solutions (Coelho et al. 2021, Dauzère-Pérès et al. 2023, Jiang et al. 2023). Among others, Cheng et al. (2016) have presented a combined approach for the multi-objective flexible multi-purpose workshop scheduling problem. Computational results demonstrate that the proposed algorithm is an effective approach for multi-objective FJSP, particularly for large-scale problems.

Kaplanoğlu (2016) proposed an Object-Oriented (OO) approach for the multi-objective FJSP along with a simulated annealing optimization algorithm. Most solution approaches in the literature typically use a two-row coding scheme to represent this problem. However, OO analysis with design and programming methods in an encoder program helps to solve this problem. Wu et al. (2017) focused on the impact of providing different chromosomes in developing GA algorithms for solving the Flexible Job Shop Scheduling Problem (DFJSP). DFJSP is a highly NP-Hard problem, and most previous studies have developed GA algorithms to tackle it. They introduced a new chromosome representation and a novel genetic algorithm named GA_OP for solving the DFJSP. Computational results show that their proposed GA algorithm outperforms previous approaches.

The present paper also deals with addressing environmental impacts leading to reducing resource consumption. Such a stream of research has gained significant attention in recent years (Alvarez-Meaza et al. 2021, Zhang et al. 2023, Narayanan et al. 2023). In this regard, Dai et al. (2014) have proposed an energy-centric scheduling model in FJSP intending to minimize energy consumption and job completion time. They utilized heuristic algorithms, including GA and SA, to solve the problem and conducted a case study in a factory. Salido et al. (2016) have focused on the FJSP problem with machines of different speeds and provided analytical formulas to estimate the relationship ratio between parameters. The results indicate the existence of a clear relationship between reliability and energy efficiency, which demonstrates progress in the production scheduling status. They have concluded that obtaining energy-saving solutions facilitates the achievement of robust solutions and vice versa. Shrouf et al. (2014) utilized the GA algorithm for optimizing the scheduling of single-machine production. Their objective is to minimize the electricity costs associated with their programs. However, they focus on determining the start time of tasks and overlook the sequence time of the jobs.

Chang et al. (2014) model the Permutation Flow Shop Scheduling (PFSS) problem by adding job delay time for increased flexibility in processing time and solving their problem using the GA algorithm. The experiments demonstrate that their new PFSS model can reduce electricity costs by 16.83%, and their objective is cost-saving in the factories' electricity expenses. Energy consumption is a key concern in the manufacturing sector due to its negative environmental impact. Mansouri and Aktas (2016) proposed a Multi-Objective Genetic Algorithm (MOGA) for sequence-dependent two-machine permutation in the FJSP problem to combat energy consumption shortages. They compared their proposed algorithm with CPLEX and random search over a wide range of problem instances.

By reviewing the above studies and extending the relevant literature, we aim to develop a problem that simultaneously considers the energy aspects and approximate quantities in FJSP. We will introduce algorithms with a fuzzy search approach in line with the given structure to solve the developed problem.

3. MODEL FORMULATION

Here, the problem of allocating operations to jobs is addressed while determining the sequence of tasks. In this respect, the objective is to find a suitable allocation and sequence in a way that optimizes the performance criteria. Now, since the complexity of the FJSP problem is higher than that of the JSP (and the JSP problem is known to be NP-Hard), it is probable that FJSP also falls within the category of NP-Hard problems (Meng et al. 2023).

Before model formulation, the underlying assumptions as well as the symbols and parameters need to be introduced.

The following assumptions are considered for the problems:

- Machine setup times and operation transfer times are negligible.
- There are n jobs and m types of machines.
- Jobs are independent of each other.
- Machines are independent of each other.
- At any given time, a machine can only perform one operation, and it becomes available for other jobs only after completing the previous operation on that machine.
- There are no inter-operation dependencies between different jobs.
- The required energy consumption for machines is approximately (fuzzy) determined.
- The energy consumption of machines is independent of each other.

The parameters and symbols are defined below.

i, h: Job indices $(1\ldots n)$

j, g: Operation indices $(1\ldots J_i)$

k: Machine index $(1\ldots m)$

n: Set of jobs

m: Set of machines

J_i: Total number of operations for job i

M: A large number

V: Consumption voltage

TIE: Total energy in hand

$\tilde{I}_{ijk}$: Indicates the power consumption of operation *j* in job *i* on machine k

P_{kij}: Processing time of operation O_{ij} if executed on machine *k*

C_{max}: Completion time of jobs

FTEC: Fuzzy total energy consumption

C_{ij}: Completion time of operation O_{ij}

S_{ijk}: Start time of operation O_{ij} on machine *k*

C_{ijk}: Completion time of operation O_{ij} on machine *k*

C_i: Completion time of job *i*

X_{ikj}: Equals one if operation O_{ij} is executed on machine *k*, and zero otherwise.

Z_{ijhgk}: Equals one if operation O_{ij} is a predecessor of operation O_{hg} on machine *k*, and zero otherwise

Then, the base model of the problem is presented as follows. In this model, equation 1 calculates the fuzzy energy consumption of the model. This equation essentially expresses the minimization of power consumption in the industry. Equation 2 ensures that the completion time of all jobs is greater than or equal to the completion time of each job. Equation 3 ensures that the completion time of all jobs is greater than or equal to the completion time of the final operation of each job on machine k. Equation 4 states that the start and completion time of a job on machine k is zero if operation Oij is not assigned to machine k. Equation 5 states that the difference between the start and completion times of jobs is equal to the minimum processing time on machine k. Equations 6 and 7 indicate that operations Oij and Ohg cannot be executed simultaneously on the same machine in the set Mij \ Mhg. Equation 8 ensures that the precedence relationships between operations are not violated. For example, operation Oij cannot start before Oij-1. Equation 9 ensures that operations are assigned to only one machine if operation Oij is not assigned to the machine.

$$FTEC = \sum_{i=1}^{n} \sum_{j=1}^{J_I} \sum_{K=1}^{m} \sqrt{3} * V * (X_{ijk} * \tilde{I}_{ijk}) \qquad \forall i, j, k \tag{١}$$

$$C_{\max} \geq C_i \qquad \forall i \tag{2}$$

$$C_i \geq \sum_{k \in M_{ij}} C_{ijk} \qquad \forall i, j = J_i \tag{3}$$

$$S_{ijk} + C_{ijk} \leq X_{ijk} * M \qquad \forall i, j, \forall k \in M_{ij} \tag{4}$$

$$C_{ijk} \geq S_{ijk} + P_{kij} - (1 - X_{ijk}) * M \qquad \forall i, j, \forall k \in M_{ij} \tag{5}$$

$$S_{ijk} \geq C_{hgk} - (Z_{ijhgk}) * M \qquad \forall i \leq h, \forall j, g, \forall k \in M_{ij} \cap M_{hg} \tag{6}$$

$$S_{hgk} \geq C_{ijk} - (1 - Z_{ijhgk}) * M \qquad \forall i \leq h, \forall j, g, \forall k \in M_{ij} \cap M_{hg} \tag{7}$$

$$\sum_{k \in M_{ij}} S_{ijk} \geq \sum_{k \in M_{ij}} C_{ij-1k} \qquad \forall i, \forall j = 2, \ldots, J_i \tag{8}$$

$$\sum_{k \in M_{ij}} X_{ijk} = 1 \qquad \forall i, j \tag{9}$$

$$S_{ijk} \geq 0, C_{ijk} \geq 0 \qquad \forall i, j, k \tag{10}$$

$$c_i \geq 0 \qquad \forall i \qquad (11)$$

$$X_{ijk} \in \{0,1\} \qquad \forall i, j, k \qquad (12)$$

$$Z_{ijhgk} \in \{0,1\} \qquad \forall i \leq h, \forall j, g, \forall \mathrm{k} \in \mathrm{M_{ij}} \cap \mathrm{M_{hg}} \qquad (13)$$

Equation 14 states that the sum of energy consumption must be less than or equal to the total available energy. FTEC in equation 14 represents the total fuzzy energy consumption.

$$FTEC \leq TIE \qquad (14)$$

4. Solution Methodology

Nowadays, as problems grow in complexity and the importance of finding solutions quickly increases, classical methods are no longer sufficient to handle many engineering problems. In numerous engineering problems, the search space exponentially grows with the problem's dimension, making classical methods impractical due to computational limitations. Consequently, in most cases, random search algorithms are preferred over exhaustive search methods. Evolutionary algorithms and intelligent particle algorithms have shown remarkable growth in addressing these challenges. Metaheuristic search algorithms are intelligent algorithms inspired by physical, biological, and natural processes. They intelligently explore the search space using probabilistic rules to optimize a black-box-defined problem.

4.1. Simulated Annealing Algorithm

The Simulated Annealing (SA) algorithm is one of the well-known metaheuristic algorithms in the field of artificial intelligence algorithms (Lim et al. 2023). This method, like other metaheuristic algorithms, is based on emulating and simulating a natural law or relationship. In the annealing process, metals are initially heated to a very high temperature and then gradually cooled over time. During the heating phase, the speed of atomic motion in the metal significantly increases. Subsequently, during the gradual cooling, specific patterns in the arrangement of atoms emerge. These changes in the atomic patterns lead to the development of valuable properties in the cooled metal, such as increased strength.

The SA algorithm follows a similar principle by mimicking the annealing process. It begins with an initial solution and performs iterative steps. In each step, a neighboring solution is formed, and its quality is evaluated. If the neighboring solution is better than the current solution or meets certain probability conditions, it becomes the new current solution. Otherwise, the algorithm may explore other neighboring solutions. Throughout the iterations, the algorithm gradually decreases the temperature and updates its parameters to guide the search towards an optimal solution. The SA algorithm is depicted in Fig.1, and its steps are outlined in the algorithm flowchart. The algorithm's initialization involves providing problem-related information and adjusting its parameters, such as initial temperature, cooling rate, and stopping criteria.

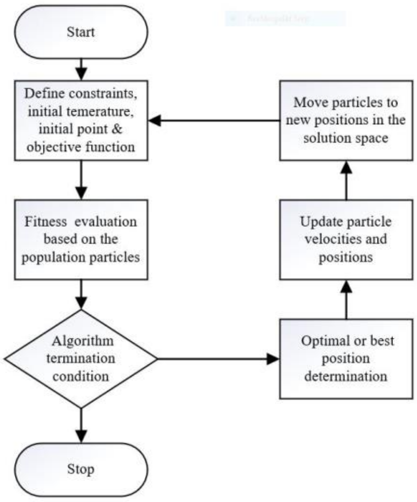

Fig ١.. Simulated Annealing algorithm

In summary, the Simulated Annealing algorithm is an effective optimization technique that is inspired by the annealing process in metallurgy. It has been widely applied to various real-world optimization problems, providing valuable solutions by intelligently exploring the solution space and converging towards better solutions over time.

4.2. Genetic Algorithm

The Genetic Algorithm (GA) is one of the most renowned optimization algorithms in the field of optimization and artificial intelligence. It is inspired by the principles of natural selection and genetic inheritance in living organisms. In GA, candidate solutions are represented as chromosomes and genes, and through genetic operators, such as mutation and crossover, new candidate solutions are generated by combining and modifying the genetic information. The algorithm iteratively evolves and refines these solutions, with better-fit individuals surviving and propagating to the next generations. The GA operators allow it to explore diverse regions of the solution space and converge towards better solutions efficiently. In this domain, GA can fine-tune molecular structures to improve their binding affinity with target proteins, leading to increased therapeutic efficacy. GA's ability to handle complex solution spaces and adapt to changing environments, resembling the natural evolution process, makes it a valuable optimization tool in various scientific and engineering fields, including biology and chemistry.

The Genetic Algorithm (GA) is a well-established optimization technique wherein each candidate solution is represented as an individual chromosome. Each chromosome possesses its fitness value, and a higher fitness corresponds to a better solution. In GA, there are also other typical operators with whom one could acquire the required knowledge by following the related studies (Gen and Lin 2023, Nicholl 2023). The general procedural framework of the GA algorithm involves generating an initial population of individuals randomly without any specific selection criteria. Then, solutions are evaluated and sorted while eliminating redundant members from the population if they exist. Selecting a set of the fittest individuals in the population as parents based on their fitness levels besides performing crossover operations among them to create a population of offspring are the next steps. Then, it comes to randomly selecting individuals from the population, applying mutation operations to them, and generating a population of mutated individuals. The original population, the population of offspring, and the population of mutated individuals are further merged. This process is repeated until the termination condition is fulfilled.

5. RESULT

In this section, the results associated with GA and SA algorithms are discussed and the necessary comparison is conducted.

5.1. Problem-solving by genetic algorithm

A standard job shop schedule can minimize the makespan time by Genetic algorithm in both single and multi-objective problems (Kumar and Dhas, 2023). The proposed single-objective algorithm GA has been used. The algorithm is applied to 10 randomly generated distinct problems involving tasks, machines, and diverse operations. The results of these experiments are presented in Table 1. As illustrated in Fig. 2, the proposed GA algorithm demonstrates remarkable performance by obtaining optimal solutions in significantly reduced computational times as the problem size increases.

Table 1- Fuzzy consumed energy obtained from the GA and SA algorithms.

Problem	n*m*o	Iteration	Population	$GA_{(Fuzzy)}$				$SA_{(Fuzzy)}$			
				Time	Outputs			Time	Outputs		
1	4*5*3	100	100	6.85	627.55	633.89	640.23	14.74	627.55	633.89	640.23
2	4*7*4	100	100	6.54	658.41	665.06	671.71	15.69	504.10	509.19	514.28
3	7*6*3	100	100	7.23	853.88	862.50	871.13	16.72	699.56	706.63	713.70
4	7*9*5	100	100	7.35	668.70	675.45	682.21	16.69	699.56	706.63	713.70
5	8*8*5	100	100	7.37	1,100.78	1,111.90	1,123.02	19.83	853.88	862.50	871.13
6	9*7*4	100	100	7.40	823.02	831.33	839.64	6.66	915.60	924.85	934.10
7	15*7*6	100	100	9.18	1,995.81	2,015.97	2,036.13	8.34	1,851.79	1,870.49	1,889.20
8	30*11*8	100	100	11.27	3,971.05	4,011.16	4,051.27	8.28	2,057.54	2,078.32	2,099.11
9	50*16*6	100	100	14.30	4,423.71	4,468.39	4,513.08	15.10	4,341.41	4,385.26	4,429.11
10	60*25*8	100	100	18.37	4,238.53	4,281.34	4,324.16	18.08	4,228.24	4,270.95	4,313.66
		SUM		95.86	19,361.44	19,557.00	19,752.58	140.13	16,779.23	16,948.71	17,118.22

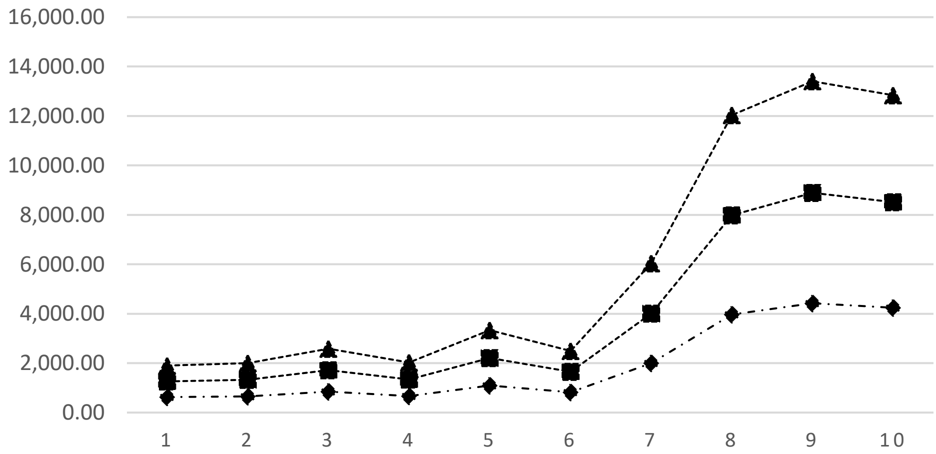

Fig. 2. GA performance

5.2. Problem-solving by simulated annealing algorithm

In this section, the proposed single-objective algorithm named SA is examined. As depicted in Fig. 3, it can be observed that as the size of the problem under consideration increases, the proposed SA algorithm obtains optimal solutions in significantly reduced computational times. It provides insights into how the simulated annealing algorithm, enhanced with fuzzy logic, performs in solving flexible workshop scheduling problems of varying dimensions. It demonstrates the algorithm's effectiveness in improving the solutions' quality as reflected in the reduction of energy consumption.

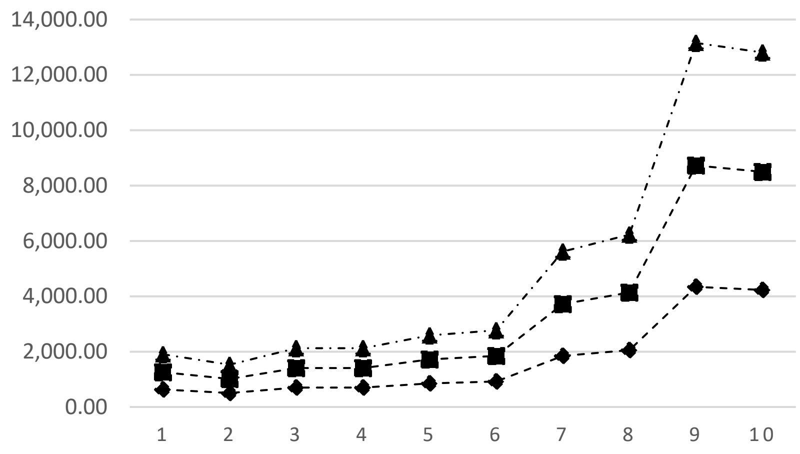

Fig. 3. SA Performance

5.3. Comparison of genetic algorithm and simulated annealing algorithm

In this section, the algorithms are compared with each other based on the content of Table 2 and the representation of Fig.4 with respect to 10 different optimization problems. It is noticeable that SA with Fuzzy Logic has exhibited a marginally superior ability to attain improved solution quality in specific optimization problems when contrasted with GA. This can be ascribed to a set of fundamental characteristics inherent to SA-Fuzzy, rendering it particularly effective in addressing intricate and demanding optimization landscapes.

Firstly, the prowess of SA-Fuzzy can be attributed to its global exploration capability. By employing a probabilistic acceptance mechanism, SA-Fuzzy can judiciously consider inferior solutions during the initial stages of the optimization process. This enables the algorithm to explore a wide spectrum of potential candidate solutions, encompassing unexplored and potentially more promising regions within the search space. Secondly, the efficacy of SA-Fuzzy stems from its adaptive temperature schedule. Through a meticulously designed cooling schedule, SA-Fuzzy adeptly regulates the acceptance probability of inferior solutions at each iterative step.

Table 2 – Defuzzied results obtained from the comparison of the algorithms

		GA		SA	
Problem	n*m*o	Time	Fuzzy Energy Consumption	Time	Fuzzy Energy Consumption
1	4*5*3	6.85	**627.55**	14.74	**627.55**
2	4*7*4	6.54	658.41	15.69	**504.10**
3	7*6*3	7.23	853.88	16.72	**699.56**
4	7*9*5	7.35	**668.70**	16.69	699.56
5	8*8*5	7.37	1,100.78	19.83	**853.88**
6	9*7*4	7.40	**823.02**	6.66	915.60
7	15*7*6	9.18	1,995.81	8.34	**1,851.79**
8	30*11*8	11.27	3,971.05	8.28	**2,057.54**
9	50*16*6	14.30	**4,423.71**	15.10	4,341.41
10	60*25*8	18.37	4,238.53	18.08	**4,228.24**
SUM		**95.86**	19,361.44	140.13	**16,779.23**

We can also observe that the time taken by GA-Fuzzy for each problem instance varies but the majority of them are lower than their counterpart in SA. Some problems are solved relatively quickly, with time values around 6 to 7 seconds. In contrast, more complex problems significantly more time, with time values of around 11 to 18 seconds. This total time represents the combined effort spent by GA-Fuzzy in exploring the search space, applying genetic operators, and iteratively improving the candidate solutions to find satisfactory or near-optimal solutions for each problem. Although the GA algorithm achieved solutions in less time compared to the SA algorithm, the superiority of the SA algorithm over the GA algorithm is noticeable for most of the problem sizes of the 10 examined cases.

6. Conclusion

Efficiently optimizing the utilization of available resources and facilities holds paramount importance in enhancing productivity, optimizing capacity allocation, minimizing task completion times, and ultimately fostering profitability for organizations. Among the prominent scheduling challenges, the FJSP stands out as a generalized version of the JSP. In this context, machines exhibit flexibility, enabling them to undertake multiple operations. As a consequence, in addition to determining the task sequencing, the allocation of machines to specific operations assumes a pivotal role, rendering the FJSP more intricate than its JSP counterpart and placing it within the realm of NP-Hard problems. To address this complex scenario, this research has developed a fuzzy FJSP model, encompassing considerations for energy utilization. The problem's resolution employed two metaheuristic algorithms, namely GA and SA, which were applied to solve the fuzzy FJSP problem. Subsequently, these algorithms were subjected to a comparative analysis. Considering the aforementioned investigations, it is evident that without employing the proposed algorithms, we would not have achieved the desirable results. This research significantly impacted energy consumption patterns and the overall cost which is a crucial topic in industrial management and industrial engineering production models. To extend the present paper, future studies can integrate the current setting with other factors, such as inventory, transportation, and location.

References

Alvarez-Meaza, I., Zarrabeitia-Bilbao, E., Rio-Belver, R. M., & Garechana-Anacabe, G. (2021). Green scheduling to achieve green manufacturing: Pursuing a research agenda by mapping science. *Technology in Society*, *67*, 101758.

Chang, H. C., Tsai, H. Te, & Liu, T. K. (2014). Application of genetic algorithm to optimize unrelated parallel machines of flexible job-shop scheduling problem. *IEEE International Conference on Control and Automation, ICCA*, 596–599. https://doi.org/10.1109/ICCA.2014.6870986

Cheng, X., Gao, F., Yan, C., Guan, X., Liu, K., Chen, S., Yao, N., & Cai, J. (2016). Permutation flow shop scheduling with delay time under time-of-use electricity tariffs. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, *2016-Septe*, 2743–2748. https://doi.org/10.1109/WCICA.2016.7578700

Coelho, P., Pinto, A., Moniz, S., & Silva, C. (2021). Thirty years of flexible job-shop scheduling: a bibliometric study. *Procedia Computer Science*, *180*, 787-796.

Dai, M., Tang, D., Zhang, H., & Yang, J. (2014). Energy-aware scheduling model and optimization for a flexible flow shop problem. *26th Chinese Control and Decision Conference, CCDC 2014*, 323–328. https://doi.org/10.1109/CCDC.2014.6852166

Dauzère-Pérès, S., Ding, J., Shen, L., & Tamssaouet, K. (2023). The flexible job shop scheduling problem: A review. *European Journal of Operational Research*.

Gen, M., & Lin, L. (2023). Genetic algorithms and their applications. In Springer handbook of engineering statistics (pp. 635-674). London: Springer London.

Jiang, B., Ma, Y., Chen, L., Huang, B., Huang, Y., & Guan, L. (2023). A Review on Intelligent Scheduling and Optimization for Flexible Job Shop. *International Journal of Control, Automation and Systems*, *21*(10), 3127-3150.

Kaplanoılu, V. (2016). An object-oriented approach for multi-objective flexible job-shop scheduling problem. *Expert Systems with Applications*, *45*, 71–84.

Kumar, K. A., & Dhas, E. R. (2023). Opposition based genetic optimization algorithm with Cauchy mutation for job shop scheduling problem. *Materials Today: Proceedings*, *72*, 3006-3011. Li, M., & Wang, G. G. (2022). A review of green shop scheduling problem. *Information Sciences*, *589*, 478-496.

Lim, K. C. W., Wong, L. P., & Chin, J. F. (2023). Simulated-annealing-based hyper-heuristic for flexible job-shop scheduling. Engineering Optimization, 55(10), 1635-1651.

Mansouri, S. A., & Aktas, E. (2016). Minimizing Energy consumption and makespan in a two-machine flowshop scheduling problem. *Journal of the Operational Research Society*, *67*(11), 1382–1394. https://doi.org/10.1057/jors.2016.4

Meng, L., Duan, P., Gao, K., Zhang, B., Zou, W., Han, Y., & Zhang, C. (2023). MIP modeling of energy-conscious FJSP and its extended problems: From simplicity to complexity. *Expert Systems with Applications*, 122594.

Narayanan, A., Pournaras, E., & Nardelli, P. H. (2023, June). Collective Learning for Energy-centric Flexible Job Shop Scheduling. In *2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)* (pp. 1-6). IEEE.

Nicholl, D. S. (2023). *An introduction to genetic engineering*. Cambridge University Press.

Salido, M. A., Escamilla, J., Barber, F., Giret, A., Tang, D., & Dai, M. (2016). Energy efficiency, robustness, and makespan optimality in job-shop scheduling problems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, *30*(3), 300–312. https://doi.org/10.1017/S0890060415000335

Shrouf, F., Ordieres-Meré, J., García-Sánchez, A., & Ortega-Mier, M. (2014). Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *Journal of Cleaner Production*, *67*, 197–207. https://doi.org/10.1016/j.jclepro.2013.12.024

Wu, M. C., Lin, C. S., Lin, C. H., & Chen, C. F. (2017). Effects of different chromosome representations in developing genetic algorithms to solve DFJS scheduling problems. *Computers and Operations Research*, *80*, 101–112. https://doi.org/10.1016/j.cor.2016.11.021

Zhang, W., Zheng, Y., & Ahmad, R. (2023). An energy-efficient multi-objective scheduling for flexible job-shop-type remanufacturing system. Journal of Manufacturing Systems, 66, 211-232.