International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6145
***Article Received**: 29 November 2022 **Revised**: 27 December 2022 **Accepted**: 19 January 2023*

Energy-Efficient Flexible Flow Shop Scheduling With Due Date and Total Flow Time

[1]Eirtty Kapoor, [2]Ashish Kumar, [3]Devendra Singh
[1]Research Scholar, IFTM University, Moradabad
[1]Kapoor.eirtty@gmail.com
[2]Professor, ITS Engineering College, Greater Noida
[2]ashishcse29@gmail.com
[3]Professor, IFTM University, Moradabad
[3]devendrasingh@iftmuniversity.ac.in

Abstract

One of the most significant optimization issues facing a manufacturing company is the flexible flow shop scheduling problem (FFSS). However, FFSS with uncertainty and energy-related elements has received little investigation. Additionally, in order to reduce overall waiting times and earliness/tardiness issues, the topic of flexible flow shop scheduling with shared due dates is researched. Using transmission line loadings and bus voltage magnitude variations, an unique severity function is formulated in this research. Optimize total energy consumption, total agreement index, and make span all at once. Many different meta-heuristics have been presented in the past to find near-optimal answers in an acceptable amount of computation time. To explore the potential for energy saving in shop floor management, a multi-level optimization technique for flexible flow shop scheduling and integrates power models for individual machines with cutting parameters optimisation into energy-efficient scheduling issues is proposed. However, it can be difficult and time-consuming to fine-tune algorithm-specific parameters for solving FFSP.

Keywords: Scheduling, Jaya Algorithm, classification, Parallel Machine Scheduling

I. INTRODUCTION

Since a good scheduling plan increases the company's productivity effectively, shop scheduling is crucial to manufacturing scheduling. The shop scheduling problem can be broken down into three distinct categories: Using parallel processors, it is possible to make a schedule for a flow shop, a job shop, or a single machine. The flow scheduling (FSP) appears to be a common issue in manufacturing. There are many different kinds of FSPs, such as the permutation FSP (PFSP), the no-wait FSP (NWFSP), and the blocking FSP (BFSP), no-idle FSP (NIFSP), non-smooth FSP (HFSP), and hybrid FSP (HFSP).

It is crucial for us to schedule our daily activities since it gives our days structure. In an airport, plane arrival and departure times must be scheduled. In schools and universities, there are set class times and exam dates. Even in the servicing industries, schedules are created. Production should be arranged similarly in the manufacturing industries. Scheduling refers to the process of assigning resources throughout time in order to finish the assigned tasks. Machines, equipment, buildings, computers, and personnel are all essential to the smooth operation of any firm. Operations management's scheduling decision-making process is crucial. Any form of industry's development depends greatly on effective scheduling. It results in increased productivity, decreased inventory, increased production efficiency, best possible resource usage, and reduced production time and cost. Any production system's efficiency could be improved by scheduling. There was discussion of various scheduling environments. One of them is HFS scheduling, which is significant. Arthanari and Ramamurthy were the ones who initially put forth the HFS scheduling problem (1971). To address the issues, they devised a branch-and-bound algorithm. Even if only one stage of a two-stage HFS problem with minimizing makespan is made up of two parallel computers, the fact that it is an NP-hard issue has been proven. It has been found that HFS scheduling is NP-hard.

Despite the fact that heuristic and meta-heuristic approaches outperform precise ones, like branch and bound, large-scale issues are intractable using exact methods. Heuristic techniques, on the other hand, are not universal and often become stopped on the path to the local optimum. Researchers are focused on problem meta-heuristic approaches that could be used to manage difficulties of varied scales in an effort to get around the shortcomings of both exact and heuristic methods. Several meta-heuristics have also been created to solve a FFSP and generate nearly optimum solutions in a reasonable amount of time. Meta-heuristics often belong to one of two broad classes of population-based algorithms. The algorithms take their cue from EAs or swarm intelligence (SIs). The General Analysis, the Exponential Smoothing, the Exponential Distribution, and the Exponential Projection are all examples of well- (EP). Well-known algorithms that use swarm intelligence include particle swarm optimization (PSO), ant colony optimization (ACO), firefly algorithm (FF), and artificial bee colony (ABC). Other population-based algorithms also include Harmony Search (HS), Bio-Geography-Based Optimization (BBO), Eco-Geography-Based Optimization (EBO), and Gravity Search (GS) algorithms. Most of these algorithms share a common characteristic, namely the need to adjust the relevant algorithm-specific parameters. GA includes settings for variables like mutation and crossover probabilities. PSO has parameters, including acceleration constants and inertia weight. The quantity of scout bees, observer bees, and employed bees is used in an artificial bee colony (ABC). Memory consideration rate and pitch adjustment rate are used in HS. Similar to this, different algorithms each have unique tuning settings. Tuning the parameters associated with an algorithm is crucial to obtaining good optimal solutions to certain challenges.

II. LITERATURE REVIEW

Scheduling mainly involves the distribution of manufacturing resources, such as labor and materials, to machines in order to carry out a number of activities with the goal of optimizing one or more performance indicators. Scheduling methods can be broken down into four categories: flow-shop scheduling (FSS), job-shop scheduling (JSS), flex-shop scheduling (FFSS), and flex-shop scheduling (JSS) are the four main categories from real-world industrial environments where scheduling issues are most commonly found (FJSS). The shop scheduling is among the most very well challenging scheduling difficulties; therefore it has attracted the attention of researchers in both academia and industry. For the purpose of tackling combinatorial optimization problems, numerous approximation techniques have been developed in the extant literature. Current approximation techniques for FFSS problems are described in this portion of the paper. Heuristic and metaheuristic techniques make up the majority of the categories for approximation methods.

Gong G., et.al. (2020) proposed that the flexible flow shop scheduling problem (FFSP) simply takes machine flexibility into account. Flexibility among employees may have a significant impact on production efficiency and output. Also, since both pollution and consumption continue to climb, manufacturers require cutting-edge methods to enhance energy efficiency. And they proposed an EFFSPW, or energy-efficient FFSP with flexible employees, in which all of these factors—machine and human adaptability, processing time, energy use, and labour costs—are considered in tandem. In order to solve the suggested EFFSPW, a hybrid evolutionary algorithm (HEA) then was developed; it featured a number of efficient operators in addition to a fresh approach to neighbourhood search variables.

Feng, et.al. (2021) conducted SI was the collective behavior of dispersed, autonomously structured natural or artificial systems. The migration patterns of monarch butterflies served as inspiration for the development of a Monarch butterfly optimization (MBO) technique, a form of high network metaheuristic algorithm. Both the migration operation as well as the butterfly modification operation are used to update MBO individuals. MBO does a better job of solving global numerical model based as well as engineering problems than many other cutting-edge optimization methods.

Zhao et.al. (2019) examined that in the manufacturing sector, the no-wait flow shop scheduling problem (NWFSP) is crucial. The broad biogeography theoretical method served as inspiration for the development of biogeography-based optimization (BBO), This employs transposition and mutation operators. In this study, we optimise the solution to the NWFSP under makespan condition by combining biogeography with variable neighbourhood search (HBV). For the purpose of creating a hypothetical starting population, we merge the NEH with its recent modifications and employ the closest neighbour technique. By combining the path relinking technique with block-based self-improvement strategy, a hybrid migration operator is created to speed up HBV's convergence. Using the iterated greedy (IG) strategy with the evolutionary algorithm in the exploitation phase improves the chances of finding a decent response.

Buddala et.al. (2018) conducted that in a simple flow shop configuration, a task with 'g' operations is completed on 'g' operation centres (stages) with only one machine per stage. There is a flexible workflow difficulty if any procedure uses many machines to provide redundancy in processing (FFSP). Due to its inclusion of both the complexity of simple flow shop problems and the complexity of scheduling parallel machines, Being NP-hard (Non-deterministic polynomially) means that it is hard to solve, FFSP has attracted a lot of attention. Given the computational difficulty of such problems, it is rarely possible to arrive at an optimal solution inside an acceptable

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6145
***Article Received**: 29 November 2022 **Revised**: 27 December 2022 **Accepted**: 19 January 2023*

length of time. In order to get close to optimal results in an acceptable length of time, numerous meta-heuristics have been presented in the past. It is a time-consuming and difficult effort to optimise algorithm-specific parameters in order to solve FFSP. For this reason, we opted to investigate two contemporary meta-heuristics, teaching-learning-based optimization (TLBO) as well as the JAYA algorithm, neither of which requires the modification of algorithm-specific parameters.

Huang et.al. (2017) investigated that to reduce energy consumption for environmental protection, concentrate on improving machine efficiency or reengineering processes. Recently, there has been lot of focus on finding ways to reduce the amount of time a machine is on in order to maximise its energy efficiency. This research takes into account three distinct machine states in order to address the issue of lowering energy bills underneath a time-of-use tariffs with really no late jobs in a flexibility flow shop, with two different speeds for processing. And suggested a hybrid genetic algorithm (GA) for solving NP-hard problems in a reasonable amount of time. The findings of this study show that, under time-of-use tariffs, energy costs can be greatly reduced without compromising on-time delivery by optimising machine states.

Rao, R. V., et.al. (2016) examined Plasma arc machining (PAM), electrodischarge machining (EDM), and micro electrodischarge machining (μ-EDM) techniques are all taken into consideration while discussing multi-objective optimization. Experiments are performed on the considered machining processes, and results are used for model building in regression analysis. It is suggested that a posteriori versions of a Jaya method be used to solve each and every multi-objective optimization methods with a single simulation run (the MO Jaya algorithm). For optimal performance in PAM, EDM, and -EDM, the MO-Jaya algorithm is used. In this study, we present a collection of Pareto-efficient solutions that may be applied to each of the machining processes that were studied.

III. CLASSIFICATION OF SCHEDULING PROBLEMS

Problems with scheduling have become increasingly complex and sophisticated. In recent years, an increase in computer-integrated manufacturing has resulted in the growth of both technology and automation, making floor shops more manageable with fewer opportunities for human error. However, the incorporation of several constraints and variables has also led in more complex scheduling issues. Since then, scheduling has been utilized in other fields with diverse floor plans and restrictions. Thus, a classification was developed that facilitates the research and evaluation of difficulties without requiring a great deal of time to identify the issue. You can classify schedules into five broad groups:

❖ **Single Machine Scheduling:** The challenge of pure sequencing with a single resource or machine and deterministic processing times is the simplest. Pure sequencing is a specific scheduling problem in which the order of the jobs controls the entire schedule. Despite its simplicity, the one-machine case is still quite significant. In a manageable model, the single-machine problem demonstrates a range of scheduling topics. The following assumptions often apply when developing models for a single machine:

- ✓ Constant Availability of Machine During Time of Scheduling.
- ✓ The machine processes jobs one at a time.
- ✓ Other job-related information is already known in beforehand. This information comprises processing time, the job's due date, and its release time.
- ✓ Job processing is non-preemptive, i.e., jobs are completed without interruption. Figure 1 depicts a schematic illustration of the scheduling of a single machine.

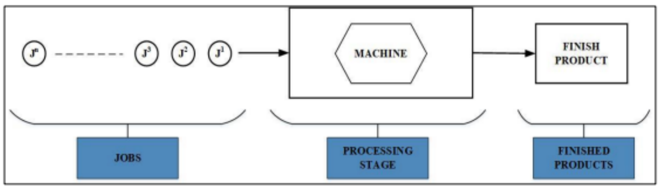

Figure 1: Representation of single machine scheduling

❖ **Parallel Machine Scheduling:** In the current industry, the presence of a parallel machine environment is commonplace. The generalized version of single machine scheduling is parallel machine scheduling. This type of configuration consists of numerous machines that are set up in parallel and are each capable of carrying out a similar set of tasks.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6145
***Article Received**: 29 November 2022 **Revised**: 27 December 2022 **Accepted**: 19 January 2023*

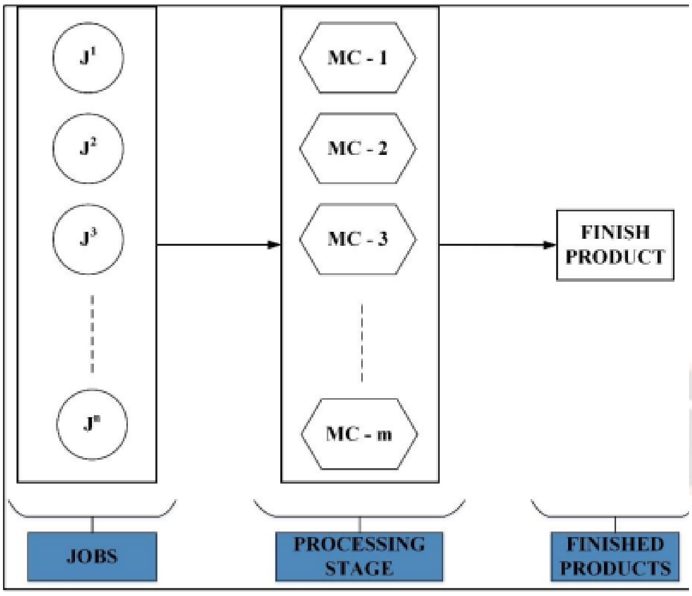

Figure 2: Representation of parallel machine scheduling

- **Flow Shop Scheduling:**Flow-shop scheduling is a term used to describe a timetable where each work in a multi-stage job processing industry follows the same path of machine visits (FSS). A visual representation of FSS is shown in Figure 3. It consists of a collection of multiple-operation jobs, ɳ-unrelated with no apparent relationship to one another that ready to be dealt with at time zero. P-operations are needed for each task, and they are carried out on several machines until the jobs are finally transformed into finished goods. With FSS, Due to technological constraints, the jobs should be transported across the devices in the same order. There are as many machines as there are stages, and there are as many activities as there are stages.

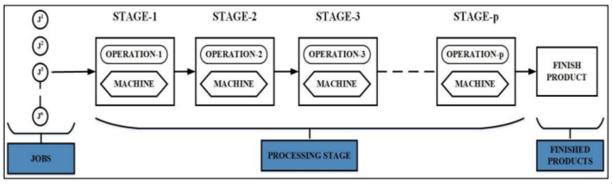

Figure 3: Representation of flow-shop scheduling

- **Job Shop Scheduling:**Job-shop scheduling is the term used when a group of jobs are to be handled in a set of machines in such a way that each job has a predetermined sequence or route of visits on the machines (JSS). A visual illustration of the JSS is presented in Figure 4. One major distinction between the fundamental JSS problem as well as the flow shop issue is that the latter does not assume a unidirectional flow of work. There really are m machines and n jobs that need to be scheduled. This is the challenge. Each job has a set of tasks that must be done that are ordered in a linear fashion, much to the flow shop concept. Although a work can have any number of operations, the most common formulation of the job shop problem requires that each task have exactly p- operations, one on each machine.

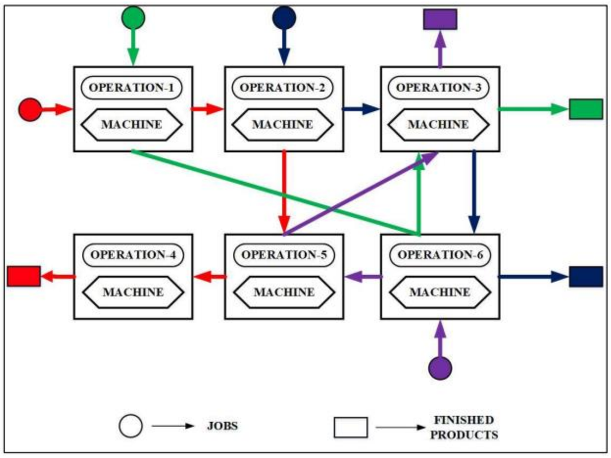

Figure 4: Representation of job-shop scheduling

- **Hybrid shop scheduling:** In a real-world environment, running into a single computer is a rare sight. It is customary in business to set up parallel machines that are identical and can do the same set of tasks at every stage of the production line. Increasing floor space utilization and reducing bottleneck formation are the two goals of this setup. A multi-stage industry that incorporates parallel machine scheduling results in hybrid shop scheduling. Flexible flow-shop scheduling (FFSS) and flexible job-shop scheduling are two subtypes of hybrid shop scheduling based on the order in which the jobs visit the various phases (FJSS). A generalized version of the traditional FSS issue, the FFSS uses identical parallel computers placed at each stage. When it comes to the FFSS problem, p-operations are represented by p-stages, and each stage consists of several machines. An illustration of the FFSS is shown in Figure 5.

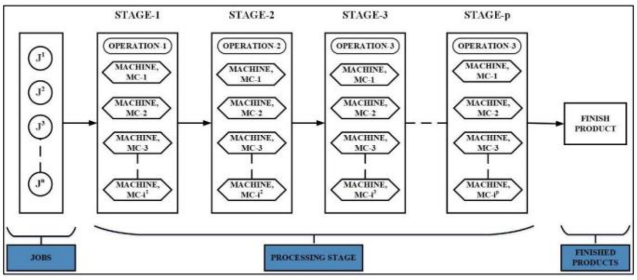

Figure 5: Representation of flexible flow-shop scheduling

IV. RESEARCH METHODOLOGY

The effective distribution of resources for the completion of activities within a predetermined timeframe is the main goal of scheduling. Typically, scheduling problems are discussed using words from the manufacturing industry like "jobs," which represent activities, and "machines," which represent resources. The scheduling problem is how to decide which jobs should be completed on which set of machines in what order to achieve one or more decision goals. The environment of a flow shop comprises of systems with a unidirectional workflow, where tasks must be completed sequentially over a number of phases. On machines 1, 2,..., m, 'n' jobs must be completed consecutively while taking into consideration comparable due dates. All machines use the same task processing routines. In the end, the minimal flow-time value decreases the system's average reaction time. By giving the shortest job to the most qualified resource, the work flow time can be shortened. It is obvious that new technology uses less energy than older technology. As a result, we should reduce early arrivals and late departures and maximize energy efficiency. Additionally, we must keep track of and control the total amount of time it takes for all tasks to be completed, also called the make span, flow rate time, and waiting time. The Jaya Algorithm and the monarch butterfly optimization algorithm are introduced as hybrid optimization techniques for resolving FFSP.

JAYA algorithm

Algorithm JAYA is suggested by Rao (2016). This strategy's core assumption is that, for each particular population, we should continuously advance towards the optimal solution while retreating from the worst. The ease of use of this method, which requires only a single equation and does not depend on adjusting any algorithm-specific parameters to achieve optimal results, makes it very straightforward to understand and makes it stand out from other meta-heuristics. JAYA just only one phase in comparison to the monarch butterfly optimization method.

The mathematical description of JAYA algorithm is as follows. Let $f(x)$ be the objective function to be optimized. At any iteration 'i', let Z_{best} and Z_{worst} denote the best and worst solutions, respectively, among the population, then a solution of the population is modified as follows:

$$Z_{new\ i}=Z_{old\ i}+r_1\times(Z_{best}-|Z_i|)-r_2\times(Z_{worst}-|Z_i|),$$

where r_1 and r_2 are the two random numbers between zero and one. The term $r_1 \times (Z_{best}-|Z_i|)$ denotes the nature of solution to move towards the best solution and the term $-r_2 \times (Z_{worst}-|Z_i|)$ denotes the nature of the solution to move away from the worst solution. The new solution is accepted if it gives a better value. The flow chart of JAYA algorithm is given in Fig. 6.

The preceding stage's completion time must be more than and equal to the earliest start time for jobs to also be processed inside the current stage, when the preceding stage's completion time is predominantly in effect. Although jobs in previous stages determine work release times for the present stage, jobs inside the current stage determine job release times for the next step. The DELAY concept is often employed to solve the scheduling issue of arbitrary job release time because It gives time to look into other possible jobs to do while a task is being done. Combining the DELAY idea to different priority rules creates new decoding methods, one of which is the DELAY(PRn) approach, where PRn is the selected priority rule.

Step 0: To begin, a list of tasks is provided, and these jobs have been given to machines in order (k=1).

Step 1: Let k=k+1. The unplanned jobs are put in ascending order based on how long it took them to finish in the previous phase, and the time it took them to finish is used as the fictional release time r'_j.

Step 2: The amount of time available is found to be t', which is the most of any of the computers at stage k.

Step 3: The candidate should set CS to "All unanticipated jobs j are found" and added to a queue CS if one's fake discharge times are below or the same as the machines' largest available time, or $r'j<t$.

Step 4: If CS={ϕ}, Let the total number of unscheduled jobs, t, be equal to $\text{Min}r'_j$. Step 3 is next.

Step 5: If CS={ϕ}, The potential jobs, jϵCS, are subject to one of the aforementioned priority rules. To schedule a job and take it out of the collection of unscheduled jobs, the top job is picked.

Step 6: Machines' current availability is updated.

Step 7: If every job is scheduled, go over to Step 8; if not, go to Step 2.

Step 8: If $k \neq m$ go to Step 1.

Step 9:C_{max} =MaxC_j.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6145
***Article Received**: 29 November 2022 **Revised**: 27 December 2022 **Accepted**: 19 January 2023*

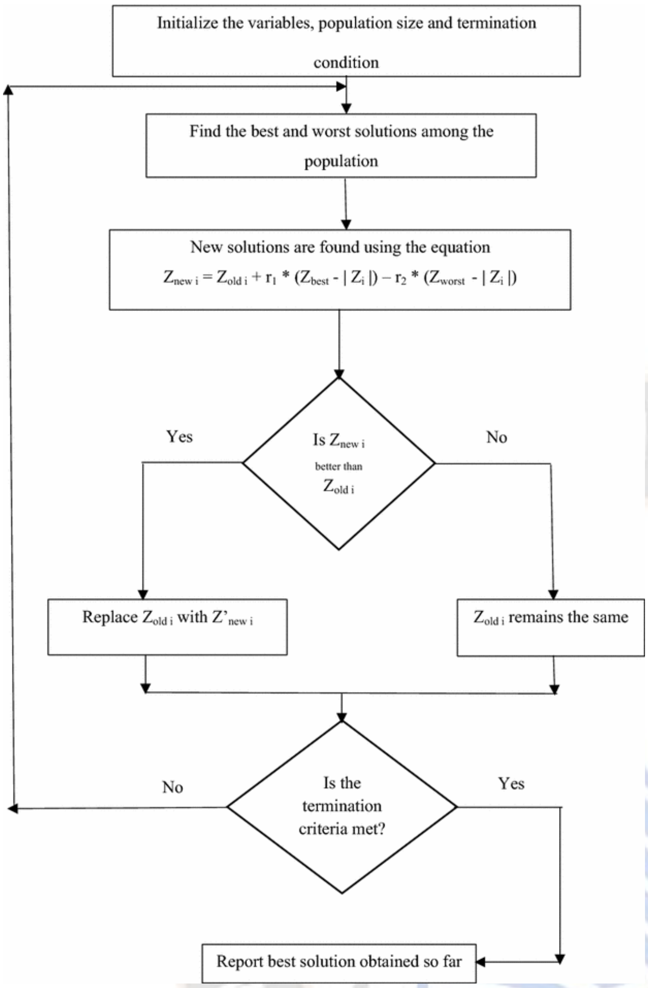

Figure 6: Flow chart of Jaya algorithm

V. RESULT

To assess the efficiency of the JAYA in solving the FFSP and Carlier, Taillard, and Reeves are frequently used benchmark datasets for comparing the efficacy of new algorithms with ones already in use. It is shown that To demonstrate how the proposed method functions, we'll examine a bus system that contains 41 power lines, 6 generator, 4 tap-changing transformers, with 2 shunt compensators.

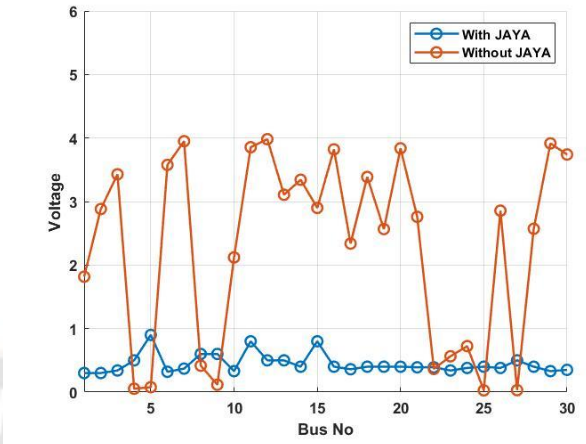

Figure 7:Voltage with bus number during Jaya

The magnitudes of a bus voltage, lines power flows, or system losses are investigated by modifying the controller parameters of the devices. the magnitude of bus voltage varies as seen in Fig. 7. Due to the fact that bus-2 is connected to the receiving end, the amplitude of the voltage variation at bus-2 is considerable.

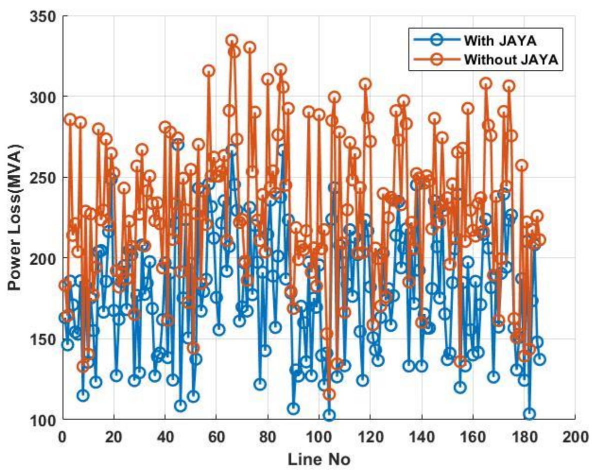

Figure 8: Power Loss with Line number During Jaya

Figure 8 depicts the variance in transmission line power loss under and without Jaya. The greatest effect on power loss is shown in the case of a simultaneous interruption of lines and generator, as shown in the accompanying diagram. Most transmission lines run closer to maximum MVA limit when there is a line outage.

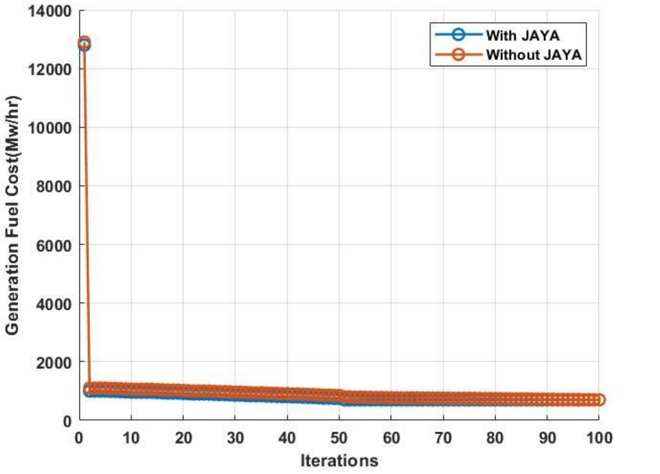

Figure 9:Iterative process with Generation during Jaya

It is calculated that the proposed Jaya method decreases the generation cost of fuel by 0.5195 $/h as compared to the prior algorithm under normal conditions. Figure 9 displays the convergence characteristics in typical settings, demonstrating that the suggested method, like the existing one, starts with a good initial value and converges to the ideal value with a smaller number of repetitions.

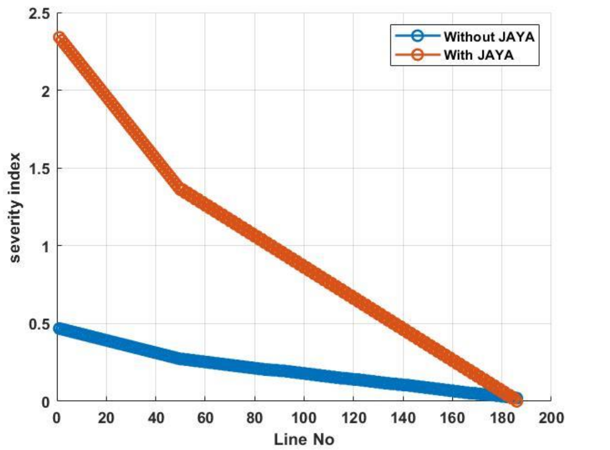

Figure 10: Severity Index with Line number During Jaya

Figure 10 depicts the convergence characteristics for with and without Jaya conditions. The number of lines shown in the figure above increases from normal to concurrent lines or generator failure situations, both for the initial value and the line count needed for final convergence.

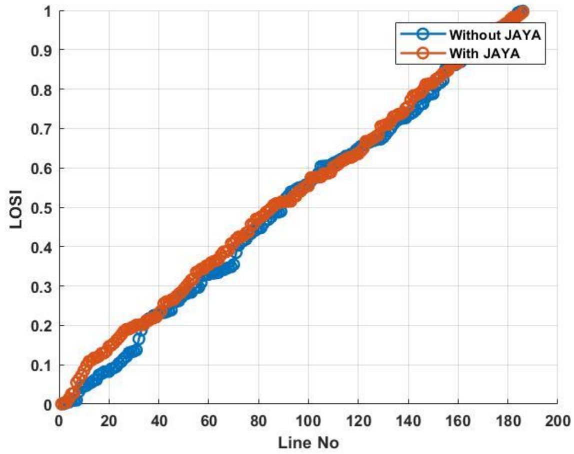

Figure 11: LOSI with Line number during Jaya

The LOSI values are determined in three different operating conditions: normal operation, peak operation, and idle operation. LOSI value variation across the system for each power line is shown in Fig. 11, but cannot be shown in full detail due to page constraints.

Table 1: Results of flexible flow shop scheduling under normal and contingency conditions

Control parameters		Normal condition	**Outage condition**		
			Lines	**Generator**	**Both lines & generator**
Real power generation (MW)	P_{G1}	74.8460	72.7799	163.5706	128.8717
	P_{G2}	75	49.9491	77.5815	-
	P_{G5}	52	52	0	0
	P_{G8}	37	37	31.7327	37
	P_{G11}	23.8406	32	18.7654	31
	P_{G13}	23.8407	21.3817	29.8113	29.2725
Generator voltages	V_{G1}	1.0369	1.08	1.0515	1.0337
	V_{G2}	1.0275	1.0601	1.0284	1.0425
	V_{G5}	1.0140	1.0287	1.0462	0.9869
	V_{G8}	1.0110	1.0318	0.9803	0.9520
	V_{G11}	1.012	1.0353	1.0138	0.9890
	V_{G13}	1.0180	1.0586	1.0280	1.0454
Generation fuel cost		924.4158	935.7120	846.8251	906.8277
Severity function value		0.3398	1.2227	1.1179	3.1236
Total power losses (MW)		4.1367	5.7611	10.4289	17.3259
Time (s)		14.5766	31.2839	23.7264	39.9485

Results for the system severity function using the suggested Jaya approach are shown in Table 1 for both baseline and emergency conditions, such as single and multiple outages of transmission lines or generators. As total generation and associated transmission power losses grow in the event of a transmission line failure, For both severity function value (0.8829) and the total generation fuel cost (11.298 $/h) increase from the baseline situation. Another thing to keep in mind is that active power generation and, consequently, transmission power loss are larger than they would be under normal conditions when a generator goes down.

VI. Conclusion

The goal of this study was to develop a multi-level optimization technique for reducing a flexible flow shop's total energy usage and production time. The cutting forces of every machine play a role in the multi-level optimization approach, which in turn affects the processing energy and time usage. This is in contrast to traditional scheduling approaches, where the amount of time spent on each task at each step is fixed in advance. Findings from the scheduling show that the multilayer optimization approach can help businesses cut down on production time and overall energy use. Synergistic energy savings gains can be gained when optimization is conducted at both the machine tool and shop floor levels. It should be mentioned that the multi-level optimization's sequential technique may result in local optimum. It is clear from the findings of the multi-level optimization strategy, though, that there is room for energy savings. The suggested approach is the initial step in integrating energy concerns at various levels.

After evaluating the system's security under emergency scenarios with ideal power loss, it was found that the existence of FFSS enhanced the system's safety. To enhance system security with this device in ideal position, LOSI analysis has been used to describe how to incorporate NR load flows into the voltage source based power injection model used by FFSS.

References

[1]. Gong, G., Chiong, R., Deng, Q., Han, W., Zhang, L., Lin, W., & Li, K. (2020). Energy-efficient flexible flow shop scheduling with worker flexibility. Expert Systems with Applications, 141, 112902.

[2]. Feng, Y., Deb, S., Wang, G. G., &Alavi, A. H. (2021). Monarch butterfly optimization: a comprehensive review. Expert Systems with Applications, 168, 114418.

[3]. Zhao, F., Qin, S., Zhang, Y., Ma, W., Zhang, C., & Song, H. (2019). A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem. Expert Systems with Applications, 126, 321-339.

[4]. Buddala, R., &Mahapatra, S. S. (2018). Improved teaching–learning-based and JAYA optimization algorithms for solving flexible flow shop scheduling problems. Journal of Industrial Engineering International, 14(3), 555-570.

[5]. Huang, R. H., Yu, S. C., & Chen, P. H. (2017). Energy-saving scheduling in a flexible flow shop using a hybrid genetic algorithm. Journal of Environmental Protection, 8(10), 1037-1056.

[6]. Rao, R. V., Rai, D. P., Ramkumar, J., &Balic, J. (2016). A new multi-objective Jaya algorithm for optimization of modern machining processes. Advances in Production Engineering & Management, 11(4).

[7]. Rezaeian, J., Derakhshan, N., Mahdavi, I., & AlizadehForoutan, R. (2021). Due dateassignment and JIT scheduling problem in blocking hybrid flow shop robotic cells withmultiple robots and batch delivery cost. International journal of industrialmathematics, 13(2), 145-162.

[8]. Zheng, J., Wang, L.,; Wang, J. J. (2020). A cooperative coevolution algorithm formulti-objective fuzzy distributed hybrid flow shop. Knowledge-Based Systems, 194,105536.

[9]. Cai, J., Zhou, R.; Lei, D. (2020). Fuzzy distributed two-stage hybrid flow shopscheduling problem with setup time: collaborative variable search. Journal of Intelligent; Fuzzy Systems, 38(3), 3189-3199.

[10]. Lin, Y. K.; Huang, D. H. (2020). Reliability analysis for a hybrid flow shop with duedate consideration. Reliability Engineering; System Safety, 199, 105905.

[11]. Schulz, S., Buscher, U.; Shen, L. (2020). Multi-objective hybrid flow shop schedulingwith variable discrete production speed levels and time-of-use energy prices. Journal ofBusiness Economics, 90(9), 1315-1343.

[12]. Chou FD (2013) Particle swarm optimization with cocktail decoding method for hybrid flow shop scheduling problems with multiprocessor tasks. Int J Prod Econ 141(1):137–145.

[13]. Cui Z, Gu X (2015) An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems. Neurocomputing 148:248–259

[14]. Marichelvam MK, Geetha M (2016) Application of novel harmony search algorithm for solving hybrid flow shop scheduling problems to minimisemakespan. Int J IndSystEng 23(4):467–481

[15]. Marichelvam MK, Prabaharan T, Yang XS (2014) Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. Appl Soft Comput 19:93–101

[16]. Mousavi SM, Zandieh M, Yazdani M (2013) A simulated annealing/local search to minimize the makespan and total tardiness on a hybrid flowshop. Int J AdvManufTechnol 64(1–4):369–388

[17]. Rao R (2016) Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J IndEngComput 7(1):19–34

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v11i2s.6145
***Article Received**: 29 November 2022 **Revised**: 27 December 2022 **Accepted**: 19 January 2023*

[18]. Singh MR, Mahapatra SS (2012) A swarm optimization approach for flexible flow shop scheduling with multiprocessor tasks. Int J Adv Manuf. Technol 62(1–4):267–277

[19]. Zhang, R., &Chiong. R. (2016). Solving the energy-efficient job shop schedulingproblem: A multi-objective genetic algorithm with enhanced local search forminimizing the total weighted tardiness and total energy consumption. Journalof Cleaner Production, 112, 3361-3375.

[20]. Zhang, Z. W., Wu, L. H., Peng, T., &Jia, S. (2019). An improved scheduling approachfor minimizing total energy consumption and makespan in a flexible job shopenvironment. Sustainability, 11. 1-21.

[21]. Zhao, F. Q, Qin, S., Yang, G. Q., Ma, W. M. Zhang, C., & Song, H. B. (2019). A factorial based particle swarm optimization with a population adaptation mechanism for the no-wait flow shop scheduling problem with the makespan objective. ExpertSystems with Applications, 126, 41-53.

[22]. Zhao, F. Q. Qin, S. Zhang, Y. Ma, W. M. Zhang, C. & Song, H. B. (2019). A hy- brid biogeography-based optimization with variable neighborhood search mech- anism for no-wait flow shop scheduling problem. Expert Systems with Applica tions, 126, 321-339.

[23]. Wu, X., Shen, X., & Cui, Q. (2018). Multi-objective flexible flow shop scheduling problem considering variable processing time due to renewable energy. Sustain- ability, 10, 841.

[24]. Xu, Y. Wang, L. Wang, S. Y. & Liu, M. (2014). An effective hybrid immune algo- rithm for solving the distributed permutation flow-shop scheduling problem. Engineering Optimization, 46, 1269-1283.

[25]. Yin, L. Li, X. Gao, L. Lu, C., & Zhang, Z. (2017). Energy-efficient job shop scheduling problem with variable spindle speed using a novel multi-objective algorithm. Advances in Mechanical Engineering 9, 1-21.

[26]. Yuan, Y. &Xu, H. (2015). Multiobjective flexible job shop scheduling using memetic algorithms.IEEE Transactions on Automation Science and Engineering. 12, 336-353.

[27]. Tirkolaee, E. B., Goli, A.; Weber, G. W. (2020). Fuzzy mathematical programmingand self-adaptive artificial fish swarm algorithm for just-in-time energy-aware flow shopscheduling problem with outsourcing option. IEEE transactions on fuzzy systems, 28(11),2772-2783.

[28]. Jiang, W.; Wu, L. (2021). Flow shop optimization of hybrid make-to-order and make-to-stock in precast concrete component production. Journal of Cleaner Production, 297,126708.