Decision Making: Applications in Management and Engineering
Vol. 5, Issue 2, 2022, pp. 176-200.
ISSN: 2560-6018
eISSN: 2620-0104
DOI: https://doi.org/10.31181/dmame0316102022n

JOB SHOP SCHEDULING USING HEURISTICS THROUGH PYTHON PROGRAMMING AND EXCEL INTERFACE

Padmanabhan Sowmia Narayanan[1], Nitish Shankar Kumar[1], Raghuram Potluru[2] and Thenarasu Mohanavelu[1]*

[1] Department of Mechanical Engineering, Amrita School of Engineering, India
[2] Maryland Health Benefit Exchange, Baltimore, USA

Received: 4 July 2022;
Accepted: 29 September 2022;
Available online: 6 October 2022.

Original scientific paper

Abstract: *Job shop scheduling problem (JSSP) has remained a challenge both for the practitioners and the researchers. A JSSP consists of multiple number of machines (m) and jobs (n). As the number of jobs increases, the complexity of the problem increases exponentially and it becomes difficult to schedule manually. Many papers in the literature discuss heuristic and metaheuristic solutions to solve Job shop scheduling problems. But there is no ease of use for practitioners who rely on their experience to schedule jobs in ad hoc sessions resulting in inefficient allocation of jobs and machines. In this paper, a job shop scheduling problem under static and dynamic conditions is solved using heuristic approaches using python programming with an MS Excel user interface. For a supplier of automotive parts with a set of jobs and machines, priority dispatching rules, viz., Shortest Processing Time (SPT), Earliest Due Date (EDD), First-In First-Out (FIFO), Critical Ratio (CR) and Slack Per Remaining Operation (S/RO) are evaluated. The obtained performance metrics such as makespan, and tardiness are compared between the heuristics to select an optimal schedule by the job shop. The user inputs the jobs, machines, start and due dates through the MS Excel interface and obtains faster, practically usable results. This reduces the time taken for job scheduling and helps in making faster productivity-based decisions to maximize resource utilization and the total time to produce the product. Integrating Python at the backend and Excel at the front end will encourage many MSMEs to perform optimized scheduling using heuristics thereby reducing the throughput time.*

Key words: *Job shop scheduling problem, priority dispatching rules, python programming, micro small and medium enterprises.*

* Corresponding author
E-mail addresses: padmanabhanpadhu2024@gmail.com(Padmanabhan.S), nitsk15@gmail.com(Nitish.S), raghuram.mechanical@gmail.com(Raghuram.P), m_thenarasu@cb.amrita.edu (Thenarasu. M)

1. Introduction

A job shop scheduling problem is a complex combinatorial optimization problem that needs a practically usable solution for MSMEs (Amaro, 2022). A lean and flexible operations can be underutilized with an ineffective scheduling process. A typical job shop consists of multiple jobs (n) and machines (m) with varying routes and different setup, and processing times. Each machine can work on only one job at a time and each job is processed in a particular order (Bakuli, 2006). The objective of the job shop scheduling is to find the best order of the jobs and operations, keeping in mind the varied routing requirements. Exact algorithms used to solve JSSPs make it necessary to find a precise algorithm to optimize the problem having finite instances. Job shop scheduling belongs to a class of NP-hard problems that are non-deterministic polynomial (Lenstra et al., 1977). This class of problems do not have an exact algorithm to find the solution in definite polynomial time. Therefore, large-scale application of these methods found it impractical in many cases because of its high computational time (Vinod and Sridharan, 2011). Because of the deficiency in computational power for solving large-scale operations, it was recognized that precise or exact methods are ineffective, therefore research for obtaining approximate solutions based on heuristic methods was developed. In today’s production environment, delivery time is critical to dealing with market competitive pressures which means industries have to deliver a wide range of products within expected delivery dates, as failure to meet deadlines can lead to loss of customers and markets. While dealing with planning problems in actual production shop floors, uncertainties hinder the use of rules based on ideal assumptions (Romero-Silva, 2022). Considering the prevailing production problems and supply uncertainties, proper scheduling has to be done, especially for small and medium enterprises (SMEs). For example, uncertainties like the lack of resources and equipment availability leads to an increase in tardiness of jobs. With disruptions in a production environment such as processing overtime or ahead, an emergency order to join, and inaccurate processing time estimates, etc., the planned schedule becomes obsolete in an actual production scenario (Raghuram and Harishankar, 2021; Cowling and Johansson, 2002). Many of the researchers consider only static JSSPs, which are impractical as job flow is dynamic in practice (Wang et al. 2019). As a result, we must constantly adjust the scheduling plan in response to changes in real conditions, which is referred to as dynamic scheduling. Fulfilling customer demands responsively while scheduling is crucial in retaining customers (Raghuram and Saleeshya, 2021). Dynamic scheduling, on the other hand, is clearly more complex and difficult to solve. The dynamic events are classified into four categories, job related, machine related, process related events and other occurrences (Suresh and Chaudhuri, 1993). As it is important to consider practical dynamic conditions, this research paper uses heuristics with a practical interface to solve both static and dynamic JSSPs.

2. Literature Review

Production scheduling is the allocation of limited production facilities such as labor, machinery, and tools to complete a variety of activities (Jiang and Zhang, 2018). Job Shop Scheduling Problem (JSSP) is one of the most essential manufacturing problems (Asadzadeh, 2015) because of its impact on overall firm and supply chain productivity. The JSSP entails sequencing a series of tasks, each with its own chain of processes, to be performed in given machines for a certain amount of time. According to varied production settings, there are various types of shop scheduling patterns, which may be further classified as single-objective or multi-objective (Xiong et al. 2022). Due to the great complexity of job-shop setups, finding a perfect solution to these problems in a reasonable length of time is difficult. JSSPs are classified as NP-hard, due to the combinatorial growth of effective options (Ghedira and Ennigrou, 2000; Garey and Johnson, 1979). As a result, there are several techniques and strategies for dealing with JSSP, and each has a distinct and direct impact on the quantity, regularity, and severity of information exchange in the shop, as well as the scheduling quality. The scheduling parameters are often evaluated using criteria such as due date sensitivity, operating costs, and setup durations (Kim and Bobrowski, 1994). Under these wide range of problems, precise or optimum approaches that yield optimum answers may take much longer to estimate, but approximation methods yield near-ideal solutions in less time (Liaqait et al. 2021). According to Delgoshaei et al. (2021), job shop scheduling is the arrangement of resources available to optimize given performance measures. The scheduling framework includes a succession of jobs and units while mediating an optimum solution job sequence on each machine under specified limitations. The shop scheduling becomes increasingly complex to solve when several performance measurements are taken into account (Admi Syarif et al. 2021). There are several approaches for solving JSSP as described in the following section.

2.1 Approaches to Solving Job Shop Scheduling

To solve scheduling problems, various methods can be used. These methods are divided into two groupings: exact and approximate methods. In JSSP, the exact methods cannot produce solutions for large-scale problems (Fox and Smith, 1984). Hence heuristic methods that return approximate solutions are used. The time required to produce approximate solutions is always less than the time required to reach exact solutions (Bulbul and Kaminsky, 2013). In the real world, obtaining an optimal solution is practically impossible. Therefore, obtaining a high-quality approximate solution will be satisfactory (Kapanoglu and Alikalfa, 2011). Hence, researchers focus on developing heuristics algorithms that can produce solutions close to optimal solutions in the least possible time. By converting production scheduling problems to equality or inequality constraints, approximate methods create one or more optimization models of the target function to arrive at an optimal solution. Most of the methods proposed in extant literature are hybrid methods, that is a combination of two or more best performing rules that were previously developed. The aim of scheduling using heuristics is to optimize the value of performance measures. Approximate methods can also be used in scheduling dynamic JSSP (Gupta and Sivakumar, 2006).

The complexity of the problem increases as the number of machines and flow sequences in the scheduling problem grow. The feasible solution grows exponentially, and approximate methods can find an optimal solution in a reasonable amount of time. As a result, approximate methods can be used to solve practical problems. Usually in small-scale industries the jobs arrive dynamically and each set of jobs have different due dates. The jobs have to be prioritized and completed before the corresponding due dates with the existing set of jobs. In the following sections we review the priority dispatching rules and describe a practical JSSP scenario from a small-scale industry.

2.1.1 Metaheuristics in solving JSSP

There are numerous papers that offer different methodologies and solutions to Job shop scheduling problems using metaheuristics. Optimization conceptual researchers are researching optimization methods based on nature that could be used as optimization techniques for engineering problems. Uniyal et al. (2022) presented an overview of the most intriguing class, the nature-inspired optimization algorithms that evolved over time and with inspiration from nature. Optimization-based procedures and approaches could help to expand, develop, and generate appropriate designs and operations (Kumar et al. 2021a). Kumar et al. (2022) have provided an in-depth examination of the most widely used and explored meta-heuristic optimization methods and nature-inspired algorithms. These have wider practical applications and remain a popular research topic and an efficient tool for solving complex optimization problems.

Meloni et al. (2004) proposed an alternative graph solution algorithm for a general formulation of the JSSP in presence of blocking and/or no-wait constraints. Artificial ants are defined such that they are easily modifiable to include new constraints and can be reconfigured for multi-objective cases. Kahraman, (2006) proposed a modified Ant Colony algorithm to solve JSSP in an acceptable amount of time. To assess the system's reliability when the available information is uncertain, use of fuzzy reliability function will be useful (Chaube et al. 2018). Pongchairerks (2019) proposed a novel two-level viz. upper and lower levels, metaheuristic algorithm to solve JSSP. The former is a population-based algorithm that acts as a parameter controller for the latter, whereas the latter seeks optimality. Kumar et al. (2021b) worked to minimize the cost while satisfying the system's availability constraints. They focused on increasing the operational time of the individual components of a system to maintain higher system reliability and improve productivity and profit by the application of nature-inspired optimization techniques such as Grey Wolf Optimization (GWO) and the Cuckoo Search Algorithm (CSA).

Uniyal et al. (2020) reviewed nature-inspired optimization along with a background of fundamentals, classification, and their reliability applications. They also demonstrate the difference between multi-objective optimization and single-objective optimization. The article provides a foundation for a few nature-inspired optimization techniques and their reliability applications. Negi et al. (2021a) provide an up-to-date review of the GWO algorithm and its usefulness in more complex real-world problem-solving. JIT-JSS, a variant of the job-shop scheduling problem in which each operation has a distinct due date was studied by Ahmadian et al. (2021). In this method, any deviation of the operation completion time from its due date incurs an earliness or tardiness penalty. The authors solved this using a variable neighbourhood search (VNS) algorithm. A JSSP with blocking (BJSS) constraints was provided by Pranzo and Pacciarelli, (2016). Blocking constraints simulated the absence of buffers (zero buffer), whereas buffers had infinite capacity in the traditional job shop scheduling model.

Particle swarm optimization (PSO) has gained popularity as one of the most popular algorithms for solving JSSP. Researchers have attempted to improve this algorithm by introducing hybrid methods. Pant et al. (2017) presented a new and improved particle swarm optimization algorithm, abbreviated MPSO, for both constrained and unconstrained nonlinear optimization problems. Negi et al. (2021b) proposed a framework for implementing a hybrid PSO-GWO algorithm (HPSOGWO) for solving reliability allocation and optimization problems in a space capsule's

Complex bridge system and Life support system. In the current research work, heuristic methods are considered and will be discussed in the next section.

2.1.2 Priority Dispatching Rules

The Priority Dispatching Rules (PDRs) are used to prioritize work in a job shop in order to improve performance measures (Thenarasu et al. 2022). Sels et al. (2012) have discussed various priority rules for JSSP which are compared and validated using different objective functions. The ranking of priority rules is checked applying it to larger problems, on the extension of multiple machines per job as well as on the introduction of sequence-dependent setup times. Dynamic arrival of jobs was also tested for the ranking of the priority rules. There are numerous papers that give an insight on what and how, JSSP with various constraints, have been solved with the help of heuristics and improved techniques, over the course of time. Kalita et al. (2016) worked on providing a heuristic approach for determining the best machine loading sequence while minimizing makespan and other performance measures. Abbas et al. (2016) employed heuristics such as shortest processing time (SPT) and longest processing time (LPT) with no delays and incorporating fatigue was employed and studied. They used a case study with a variety of jobs with different production sequences using time and motion studies and found that SPT rule provides lower makespan values when no scheduled breaks occur, while LPT performs better for scheduled breaks. Snyman and Bekker (2019) applied various dispatching rules for a dynamic JSSP. A simulation model of an auto ancillary unit to prioritize jobs, using an Analytic Hierarchy Process (AHP) based priority rules in a press shop was developed and designed (Mohanavelu et al. 2017). Thenarasu et al. (2019), also proposed using an Arena simulation model to evaluate performance measures, integrating PDRs and Multi-Criteria Decision-Making (MCDM) with Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) approach (Thenarasu et al. 2020). Further, Ashwin et al. (2022) utilized LEKIN software to analyze, compare and evaluate various PDRs to improve performance measures.

2.2 Identified Research Gaps

Two major research gaps were identified in the extant literature and industry. Firstly, solving static JSSP problems do not offer practically viable solutions to the industry. Over the years, researchers have been attempting to find different approaches to solve JSSPs. The intricacies of JSSP make it difficult to develop an appropriate and effective technique. An effective strategy that produces an improved performance for a given job shop configuration cannot generate the same outcome with another configuration. Secondly, many of the researchers use simulation and optimization algorithms solved using software, that are not affordable by MSMEs. Currently, most of the industries do not own any scheduling software because of the high licensing costs involved, and the difficulty in learning and using it on a daily basis. So, it becomes difficult for the operations team to develop effective and optimal schedules.

3. Description of Case Study and Problem Statement

Small and medium scale industries use manual scheduling which is not effective way of scheduling jobs. Commercial scheduling software available in the market are costly and not affordable by SMEs. Hence most of these companies perform scheduling operations based on experience.

Table 1. Jobs with Machine Sequence and Processing times

Jobs	Operations	Machines (m/c)	Processing time (s)
Terminal	Turning	Lathe	360
	Facing	DRO	180
	Drilling	Drill	60
	Taping	Taper	60
Shutter Body	Turning	Lathe	720
	Counter Boring	DRO	300
	Drilling	Drill	180
	Taping	Taper	90
Top cover	Facing	DRO	180
	Drilling	Drill	240
	Taping	Taper	300
Cover	Turning	Lathe	240
	Drilling	Drill	360
Pole Shoe	Cropping	Mechanical Press	20
	Shot Blasting	Shot Blast	900
	Sizing	Hydraulic Press	10
	Extrusion	Hydraulic Press	15
	Annealing	Furnace	7200
	Clipping	Mechanical Press	10
	Deburring	Vibratory Deburrer	300
	Bending	Hydraulic Press	15
	Drilling	Drill	25
	Chamfering	Chamfer	10
	Threading	Thread	10
Flange	Turning (OD)	Lathe	70
	Turning (ID)	Lathe	30
Yoke	Cutting	Mechanical Press	300
	Deburring	Vibratory Deburrer	120
	Turning	Lathe	40
	Chamfering	Chamfer	40
	Slotting	Slotting	28
	Polishing	Polishing	20
	Gimping	Gimping	66
Crank Shaft	Finishing	DRO	22
	Drilling	Drill	22
	Threading	Thread	60
	Turning	Lathe	60
Crank case	Turning	Lathe	15
	Boring	Bore	15
	Grooving	DRO	15
	Milling	Milling	75
	Drilling	Drill	75
	Chamfering	Chamfer	75
	Tapering	Taper	75

Production and supply chain risks such as machine breakdowns, lack of labor availability, lack of raw materials and arrival of new jobs also result in repeated planning and scheduling activities. The time and effort spent in scheduling makes it harder for the managers to focus on production related activities. Also, manual scheduling results in wastage of time and resources, as they cannot develop an optimized schedule. Hence, there is a need to produce an optimal schedule faster in a real-time production environment with an easy-to-use and affordable interface. A number of jobs can arrive at a shop floor within a given time horizon. Table 1 shows the different jobs that are available for scheduling on a particular day. The sequence of machines through which these jobs travel, in different paths through the machines in the shop floor are also depicted in order. The processing times of the jobs in each machine are also given.

4. Methodology

The methodology followed to solve both static and dynamic job shop scheduling problems using various heuristic methods is as follows. Data was collected from a small-scale job shop, in automotive industry, located in Chennai. A static instance of the problem was solved using five priority rules, viz., SPT, EDD, FIFO, S/RO and CR. The performance measures, viz., makespan, maximum tardiness, number of tardy jobs and total tardiness are obtained and compared to find the best solution. As jobs may be introduced after the production process has started, the problem becomes dynamic in nature. The resulting DJSSP was also solved using priority rules. Python programming was used to code the algorithms of heuristics optimization. The performance measures obtained were compared using python and the output given through MS Excel. A user interface for data collection and display of resulting schedule and performance measures was developed in MS Excel with python programming in the backend.

4.1. Solving Dynamic JSSP

Dispatching rules are used to handle job sequencing on machines. The jobs to be performed are organized for each of machines by using a job priority rule. Jobs are queued and whenever a machine is available, it must be chosen after checking which of the queued tasks will be executed on the machine. Priority rules are assigned depending on the pending jobs in order to choose the job to be processed next. Five of the PDRs are employed in this paper: SPT, EDD, FIFO, CR & S/RO.

4.1.1 Pseudo Code of Heuristic Approach

Pseudocode of heuristic approach used for prioritizing of jobs is given in this section, with comments in brackets.

Step 1: Read the Input Excel file
Step 2: Assign job, m
[*job and no of machines*]
Step 3: Create a 3D array [*Processing time of jobs, allocated machines, and a flag value are initialized to zero*]
Step 4: opr = []
Step 5: mac = []
Step 6: sum (opr)
Step 7: ddj = sum (ptj)* k

Step 8: flag1, flag2 = 0
Step 9: while flag1! = totalopr **or** flag2! = m:
Step 9(A): time.sleep(0.2)
if keyboard.is_pressed('s'):
[*Adding New Jobs*]
file_name = str(input("Enter the fileName with Ext: "))
f = open(file_name,'r')
f_read = f.readlines()
f.close()
[*Increment the No of jobs, operation array, update the due date, processing time, job completion time array*]
Step 9(B): for k in range (m):
If flag value == 0 **and** arr[i][j][0] < Dispatching Rule Cond:
[*Update the mac array by checking which jobs operation (in the queue) has the smallest processing time to be performed in that particular machine and accordingly assign that jobs operation to the machine*]
Step 9(C): var = min (i **for** i **in** mac **if** i > 0)
for k **in** range (m):
if mac[k] > 0:
mac[k] = mac[k] – var
macult[k] = macult[k] + var
Cmax = Cmax + var
[*Reduce the processing times assigned in the mac array by the min value in the mac array thus indicating that the job is being processed*]
Step 9(D): Cmax, Cseq, Cj, Um
[*Calculate the make span, job completion sequence, job completion time, mac utilization time accordingly*]
Step 10: end while
Step 11: Nj, sum (Cj), max (Tj)
[*Calculate the No of Tardy jobs, Total job completion time, Max Tardiness and other performance measures accordingly*]
Step 12: Print the performance measures thus obtained

Figure 1. constitutes diagrammatic representation of the algorithm developed. The flowchart deals with application of SPT dispatching rule for solving the JSSP problem. This flow chart can be adapted for each of the dispatching rules. The performance measures obtained are compared and the best performing rule is chosen for the given products.

4.2. MS Excel User Interface

One of the main purpose of this paper is to develop an easy to understand-and-use interface to input the data as well as to receive the required output. For this purpose, a user interface developed using MS Excel, with python programming executing the heuristics in the backend is created. It is therefore not necessary for a production scheduler to have an in-depth knowledge of either python programming or MS Excel. The excel interface consists of two input sheets and an output sheet. In the first sheet named 'Instructions', the general information, and scheduling information are collected, and instructions for data input are provided. In the second sheet, 'Scheduling', the data table is provided with various fields to input - jobs, machines, start dates, due dates. Once the data is entered, the scheduler will process the schedule using the 'Run' button to obtain the results. Figure 2. shows the 'Instructions' sheet and Figure 3. shows the 'Scheduling' sheet. The number of machines for any product can be added as needed. The program automatically recognizes the number of machines for each product and adds it to the sequence.

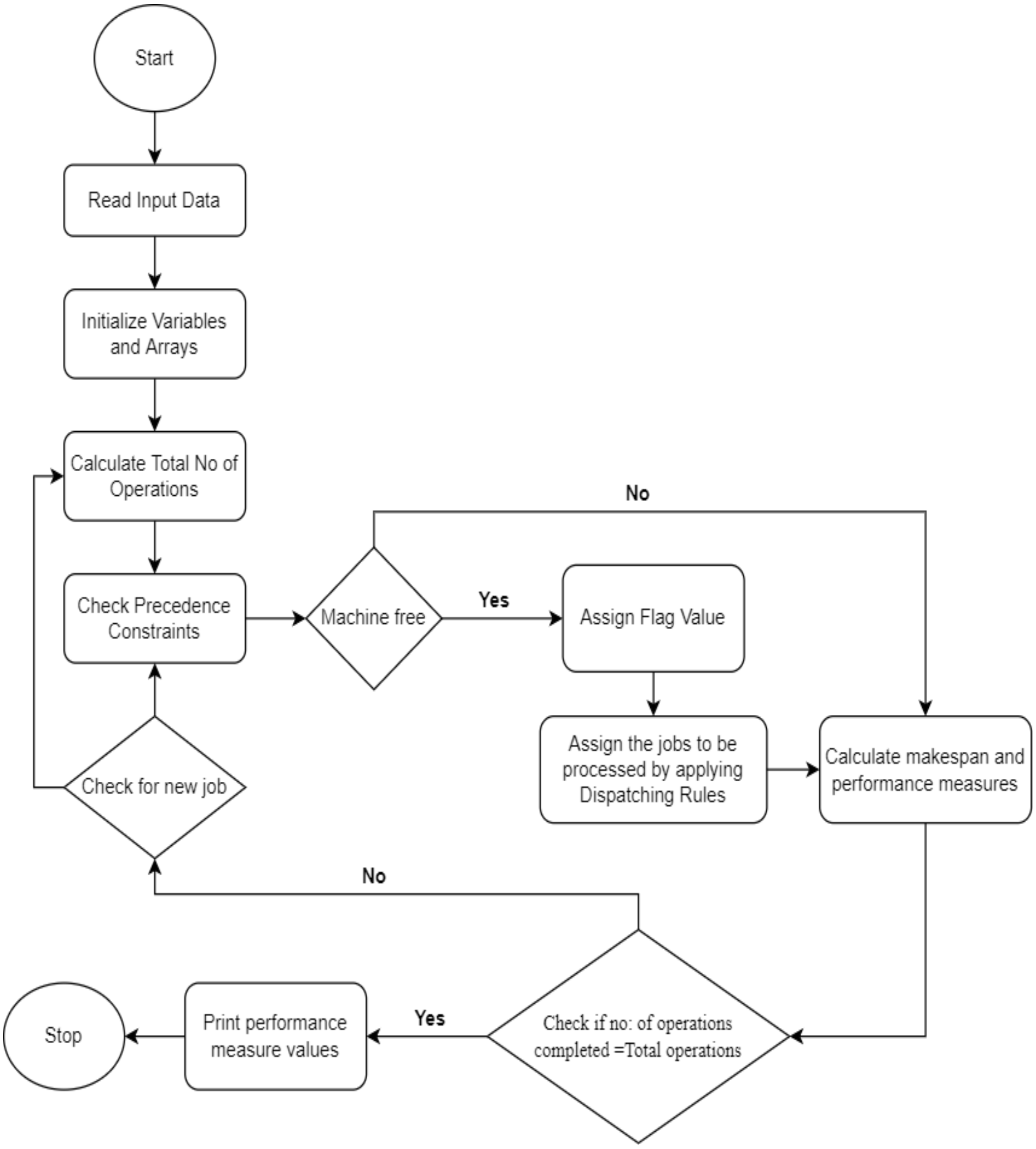

Figure 1. Flow Chart Representation of the Proposed Heuristic Algorithm

JOB SHOP SCHEDULER		
GENERAL INFORMATION		
COMPANY NAME	:	
COMPANY ADDRESS	:	
EMPLOYEE NAME	:	
EMPLOYEE ID	:	
DEPARTMENT	:	
DATE	:	
SCHEDULIG INFORMATION		
# OF JOBS	:	
# OF MACHINES	:	
SCHEDULING TYPE	:	DYNAMIC
SCHEDULING METHOD	:	HEURISTIC

INSTRUCTIONS

1. Continue to **SCHEDULING SHEET** for data input

2. Enter the data - **JOBS, OPERATIONS and MACHINES** under the rows and columns

3. If needed, insert rows and columns for a job

Figure 2. MS-Excel user interface - Information sheet

JOBS	OPERATIONS	Lathe	DRO machine	Drilling	Tapping	M. Press	H. Press	Shot Blasting	START DATE	DUE DATE
Terminal	Turning	360								
	Facing		180							
	Drilling			60						
	Tapping				60					
Shutter Body	Turning	720								
	Counter Bore		300							
	Drilling			180						
	Tapping				90					
Top Cover	facing		180							
	Drilling			240						
	Tapping				300					
Cover	Turning	240								
	Drilling			360						
Pole Shoe	Cropping					20				
	Shot Blasting							900		
	Sizing						10			
	Extrusion						15			
	Annealing									
	Clipping					10				
	Deburring									
	Bending						15			
	Drilling			25						
	Chamfer									
	Thread									

RUN

Figure 3. MS-Excel user interface - Scheduling sheet

5. Results and Discussion

The results obtained are based on the input data provided in two benchmark instances, 15x15 and 30x20 (Tasgetiren et al., 1993) and from the case company (9x16) for static instance. The total program run time taken for the heuristic methods is less than a minute for all instance. The job completion sequence, for 15x15 and 30x20 instances, using the aforementioned dispatching rules are displayed in Table 2 and Table 3 respectively. The performance measures using different heuristic methods are compared and presented in Figure. 4 Figure. 5 for 15x15 and 30x20 instances respectively.

Table 2. Job Completion Sequence (15x15) – Jobs 1-15 arranged in order of operations

SPT	8	3	12	10	7	9	2	14	1	15	5	13	6	11	4
FIFO	10	1	11	3	9	14	15	8	7	2	6	13	5	4	12
EDD	8	10	14	2	3	15	6	5	7	12	9	4	11	1	13
CR	12	10	7	2	5	14	15	8	1	9	4	6	3	13	11
S/RO	7	10	8	1	5	11	9	15	4	14	3	2	13	12	6

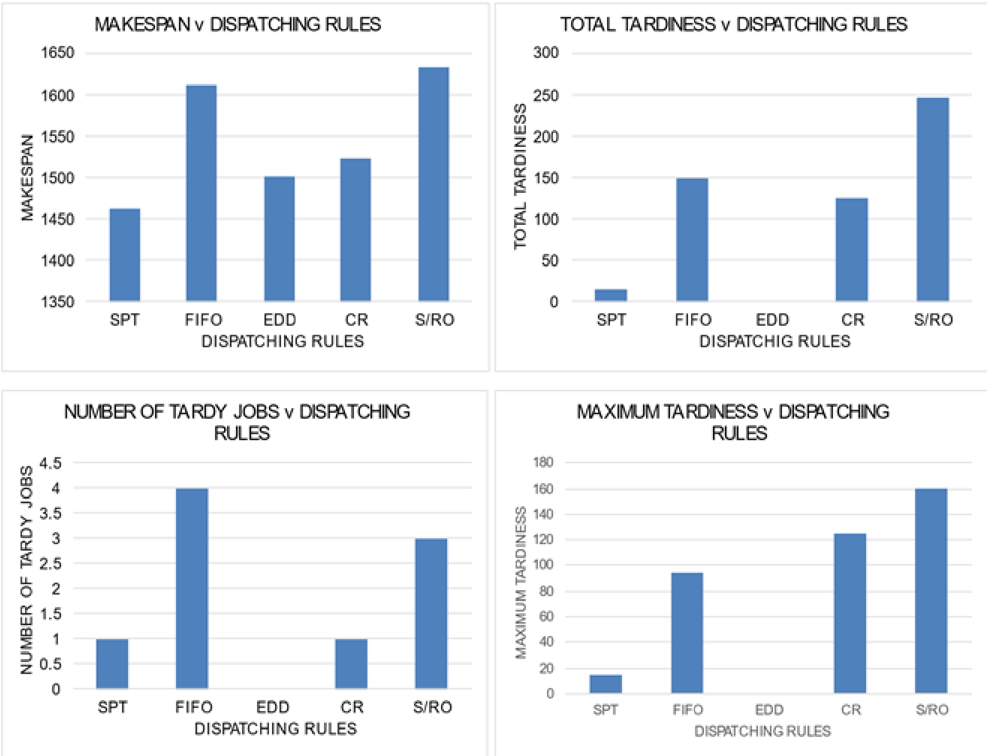

Figure 4. Graph of Performance Measures Vs. Dispatching Rules (15x15)

The simulation was carried out for the benchmark instances, 15x15 and 30x20, and the performance measures were obtained. The obtained results were cross verified and thus the code proved effective.

Table 3. Job Completion Sequence (30x20) – Jobs 1-30 arranged in order of operations

SPT	23	18	10	7	21	13	30	6	29	5	12	3	27	14	26
	2	12	17	28	25	19	11	8	24	22	1	9	16	20	4
FIFO	10	6	22	18	23	7	13	4	2	1	3	19	17	12	11
	27	5	30	16	9	14	29	26	8	24	21	28	15	20	25
EDD	6	10	7	23	27	2	14	18	9	21	29	11	22	12	4
	16	13	1	17	3	19	5	24	28	8	26	25	15	20	30
CR	11	17	26	16	4	20	3	5	13	18	9	27	19	8	1
	6	14	25	7	21	12	15	29	24	28	22	2	23	10	30
S/RO	5	23	7	18	1	19	1	13	11	25	9	15	20	16	26
	27	8	2	3	17	4	6	30	24	28	22	29	12	21	10

Figure 5. Graph of Performance Measures Vs. Dispatching Rules (30x20)

5.1 Case Study Data

The simulation was run for the data obtained from case company. There were nine different jobs, each with its comprising of 44 operations in total, to be completed within the specified due date. The job completion sequence, for 9x16 instance, using the different heuristic methods is displayed in Table 4 and the results are compared and presented in Figure. 6 for the same instance. To identify the best approach, four performance measures were taken and reviewed. From the graphical representation, it is observed that while considering makespan, SPT, FIFO and CR outperform the other dispatching rules. For Maximum tardiness, SPT has least maximum tardiness thereby outperforming the rest. For Number of Tardy jobs and Total Tardiness, SPT gives the least value making it better than the rest. Hence, for this particular problem instance, Shortest Processing Time dispatching rule is recommended.

Table 4. Job Completion Sequence (9x16) – Jobs 1-9 arranged in order of operations

SPT	6	8	3	9	4	7	1	2	5
FIFO	3	1	8	6	7	4	2	9	5
EDD	6	3	8	1	7	4	9	2	5
CR	3	1	7	8	6	4	2	9	5
S/RO	3	9	7	2	1	8	6	4	5

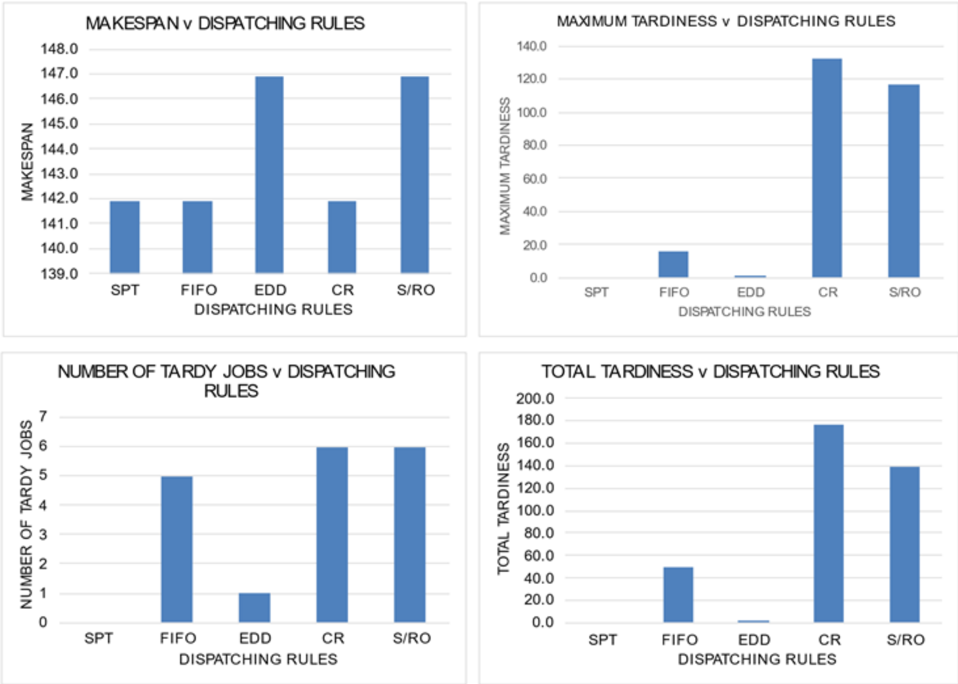

Figure 6. Graph of Performance Measures Vs. Dispatching Rules (9x16)

To validate the same, few benchmark instances were also reviewed to obtain the performance measures. To comprehend better, Table 5 to Table 7 provide the comparison of performance measures for various heuristic approaches utilized to deal with the instances 15x15, 30x20 and 9x16 respectively.

Table 5. Comparison of Dispatching rules with Performance Measures (15x15)

	MAKESPAN	MAX TARDINESS	NO. OF TARDY JOBS	TOTAL TARDINESS
SPT	**1462**	14	1	14
FIFO	1612	94	4	149
EDD	1501	**0**	**0**	**0**
CR	1524	125	1	125
S/RO	1635	160	3	246

Table 6. Comparison of Dispatching rules with Performance Measures (30x20 benchmark instance)

	MAKESPAN	MAX TARDINESS	NO. OF TARDY JOBS	TOTAL TARDINESS
SPT	**2499**	**455**	**11**	**2015**
FIFO	2529	473	18	3166
EDD	2957	669	12	2505
CR	2909	1067	21	7279
S/RO	2801	809	23	7442

For the 15x15 benchmark instance, solved using heuristics, it can be seen that SPT rule is producing better results for the performance measure, makespan. Whereas, EDD produces better results with the remaining performance measures compared to all the other heuristics. For the 30x20 benchmark instance, solved using heuristics, it can be seen that the SPT rule is producing better results with all performance measures compared to all the other heuristics.

Table 7. Comparison of Dispatching rules with Performance Measures (9x16)

	MAKESPAN	MAX TARDINESS	NO. OF TARDY JOBS	TOTAL TARDINESS
SPT	**141.9**	**0.0**	**0**	**0.0**
FIFO	**141.9**	15.6	5	50.1
EDD	146.9	1.40	1	1.4
CR	**141.9**	131.9	6	176.7
S/RO	146.9	116.9	6	139.5

With the job shop configuration, 9x16 SJSSP considered for the case company, SPT and EDD outperform the rest of the dispatching rules with least maximum tardiness. Considering makespan, SPT, FIFO and CR outperform the rest. SPT gives the least Number of Tardy jobs and Total Tardiness. Hence, for this case, Shortest Processing Time dispatching rule is recommended in solving the static job shop problem, using heuristics. It is observed that different dispatching rules affect each performance measures based on the priority given. In our case problem (9x16), SPT rule is found to be performing better for all measures because it can determine the status of specific job, establish relative priority among jobs on a common basis, relate both stock and make-to-order jobs on a common basis and also dynamically track job progress and location.

6. Conclusions

A precise schedule for static and dynamic job shop scheduling problem using a comparison of heuristic (priority rules) methods were developed using python programming. Both static and dynamic JSSP were solved for 9x16, 15x15 and 30x20 problems using priority dispatching rules, viz., SPT, EDD, FIFO, S/RO and CR. The program was run, with a case company data and benchmark problems and the following conclusions are drawn. A simple and easy to use job shop scheduler has been developed for MSMEs. It offers MSMEs, an affordable and easy-to-use solution for a time consuming, manual planning task. The above solution needs to be properly packaged for distribution to the MSME industry. As a future research work, the authors are developing an excel and python-based scheduler with metaheuristics for multiple jobs-multiple machines scenario, which can provide exact optimal solutions. It can be an effective tool in increasing the profits of the industry, as it lessens the planning time as well as improves the productivity through an optimized schedule.

6.1. Practical Implications

As there is no learning curve involved in using the program and no prior knowledge of python programming is necessary, only a basic working knowledge of MS Excel is necessary to use the interface. Dynamic jobs can be added, as in real-time, considering the current status of Work-In-Process and this is a major outcome of this research work.

Author Contributions: Methodology, Programming, writing original draft – Padmanabhan; Designing Excel Interface, writing original draft – Nitish; Conceptualization, Methodology, Supervision, writing – review & editing – Raghuram P; Supervision, Writing – review & editing – Thenarasu M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

References

Abbas, M., Abbas, A., & Khan, W. A. (2016). Scheduling job shop-A case study. In IOP Conference Series: Materials Science and Engineering. 146(1), 012052.

Admi Syarif, A. S., Pamungkas, A., Mahendra, R. K., & Gen, M. (2021). Performance Evaluation of Various Heuristic Algorithms to Solve Job Shop Scheduling Problem. International Journal of Intelligent Engineering & System, 14(2), 334-343.

Ahmadian, M. M., Salehipour, A., & Cheng, T. C. E. (2021). A meta-heuristic to solve the just-in-time job-shop scheduling problem. European Journal of Operational Research, 288(1), 14-29.

Amaro, D., Rosenkranz, M., Fitzpatrick, N., Hirano, K., & Fiorentini, M. (2022). A case study of variational quantum algorithms for a job shop scheduling problem. EPJ Quantum Technology, 9(1), 1-20.

Asadzadeh, L. (2015). A local search genetic algorithm for the job shop scheduling problem with intelligent agents. Computers & Industrial Engineering, 85(1), 376-383.

Ashwin, S., Shankaranarayanan, V., Anbuudayasankar, S. P., & Thenarasu, M. (2022). Development and Analysis of Efficient Dispatching Rules for Minimizing Flow Time and Tardiness-Based Performance Measures in a Job Shop Scheduling. In Intelligent Manufacturing and Energy Sustainability, 34(1), 337-345.

Bakuli, D. L. (2006). A survey of multi-objective scheduling techniques applied to the job shop problem (JSP). In Applications of Management Science: In Productivity, Finance, and Operations, 12, 129-143.

Bulbul, K., & Kaminsky, P. (2013). A linear programming-based method for job shop scheduling. International Journal of Scheduling, 16(2), 161-183.

Chaube, S., Singh, S. B., Pant, S., & Kumar, A. (2018). Time-dependent conflicting bifuzzy set and its applications in reliability evaluation. Advanced Mathematical Techniques in Engineering Sciences, 111-128.

Cowling, P., & Johansson, M. (2002). Using real time information for effective dynamic scheduling. European journal of operational research, 139(2), 230-244.

Delgoshaei, A., Aram, A. K., Ehsani, S., Rezanoori, A., Hanjani, S. E., Pakdel, G. H., & Shirmohamdi, F. (2021). A supervised method for scheduling multi-objective job shop systems in the presence of market uncertainties. RAIRO-Operations Research, 55, S1165-S1193.

Fox., M.S, and Smith., S.F. (1984). "ISIS—a knowledge-based system for factory scheduling," International Journal of Expert Systems, 1(1), 25–49.

Garey, M. R., & Johnson, D. S. (1979). Computers and intractability, San Francisco: freeman.

Ghedira, K., & Ennigrou, M. (2000). How to schedule a job shop problem through agent cooperation. In International Conference on Artificial Intelligence: Methodology, Systems, and Applications, Springer, Berlin, Heidelberg, 132-141.

Gupta A.K., and Sivakumar, A.I. (2006), Job shop scheduling techniques in semiconductor manufacturing, International Journal of Advanced Manufacturing Technology, 27(11), 1163–1169.

Jiang, T., & Zhang, C. (2018). Application of grey wolf optimization for solving combinatorial problems: job shop and flexible job shop scheduling cases. IEEE Access, 6, 26231-26240.

Kahraman, C. (2006). Metaheuristic techniques for job shop scheduling problem and a fuzzy ant colony optimization algorithm. In Fuzzy Applications in Industrial Engineering, 201, 401-425.

Kalita, R., Barua, P. B., & Dutta, A. K. (2016). Development of a heuristic algorithm for finding optimum machine loading sequence in fabrication shop with job shop layout. International Conference on Electrical, Electronics, and Optimization Techniques, IEEE, 1324-1329.

Kapanoglu., M and Alikalfa., M. (2011) Learning IF-THEN priority rules for dynamic job shops using genetic algorithms. Robot. Computer Integrated Manufacturing, 27(1), 47–55.

Kim, S. C., & Bobrowski, P. M. (1994). Impact of sequence-dependent setup time on job shop scheduling performance. International Journal of Production Research, 32(7), 1503-1520.

Kumar, A., Negi, G., Pant, S., Ram, M. (2021a). Availability-Cost Optimization of Butter Oil Processing System by Using Nature Inspired Optimization Algorithms. Reliability: Theory & Applications, 2(64), 188-200.

Kumar, A., Vohra, M., Pant, S., & Singh, S. K. (2021b). Optimization techniques for petroleum engineering: A brief review. International Journal of Modelling and Simulation, 41(5), 326-334.

Kumar, A., Pant, S., Ram, M. & Yadav, O. (2022). Meta-heuristic Optimization Techniques: Applications in Engineering, Berlin, Boston: De Gruyter, 1-10.

Lenstra, J.K., Rinnooy Kan, R.H.G., and Brucker, P. (1977). Complexity of machine scheduling problems, Annals of Discrete Mathematics, 1, 343-362.

Liaqait, R. A., Hamid, S., Warsi, S. S., & Khalid, A. (2021). A critical analysis of job shop scheduling in context of industry 4.0. International Journal of Sustainability, 13(14), 7684.

Meloni, C., Pacciarelli, D., & Pranzo, M. (2004). A rollout metaheuristic for job shop scheduling problems. Annals of Operations Research, 131(1), 215-235.

Mohanavelu, T., Krishnaswamy, R., & Marimuthu, P. (2017). Simulation modelling and development of analytic hierarchy process-based priority dispatching rule for a dynamic press shop. International Journal of Industrial and Systems Engineering, 27(3), 340-364.

Negi, G., Kumar, A., Pant, S., & Ram, M. (2021a). GWO: a review and applications. International Journal of System Assurance Engineering and Management, 12(1), 1-8.

Negi, G., Kumar, A., Pant, S., & Ram, M. (2021b). Optimization of complex system reliability using hybrid grey wolf optimizer. Decision Making: Applications in Management and Engineering, 4(2), 241-256.

Pant, S., Kumar, A., Bhan, S., & Ram, M. (2017). A modified particle swarm optimization algorithm for nonlinear optimization. Nonlinear Studies, 24(1),127-138.

Pongchairerks, P. (2019). A two-level metaheuristic algorithm for the job-shop scheduling problem. Complexity, 1-11.

Pranzo, M., & Pacciarelli, D. (2016). An iterated greedy metaheuristic for the blocking job shop scheduling problem. Journal of Heuristics, 22(4), 587-611.

Raghuram, P., & Harisankar,G. (2021) Modeling and Assessment of the Impact of Supply Disruption and Cost of Recovery using Systems Dynamics Approach. International Journal of Industrial and Systems Engineering, 38(4). 432-449.

Raghuram, P., & Saleeshya, P.G. (2021) Responsiveness Model of Textile Supply Chain – A Structural Equation Modeling based Investigation. International Journal of Services and Operations Management, 38(3), 419-440.

Romero-Silva, R., Santos, J., & Hurtado-Hernández, M. (2022). A conceptual framework of the applicability of production scheduling from a contingency theory approach: addressing the theory-practice gap. International Journal of Production Planning & Control, published online, 1-21.

Sels, Veronique; Gheysen, Nele; Vanhoucke, Mario (2012). A comparison of priority rules for the job shop scheduling problem under different flow time- and tardiness-related objective functions. International Journal of Production Research, 50(15), 4255–4270.

Snyman S & Bekker J. (2019), A Real-Time Scheduling System in a Sensorised Job Shop, In Proceedings of the International Conference on Competitive Manufacturing (COMA'19), Stellenbosch, South Africa,1-6.

Suresh, V., & Chaudhuri, D. (1993). Dynamic scheduling - A survey of research. International Journal of Production Economics, 32(1), 53-63.

Tasgetiren, M. F., Liang, Y. C., Sevkli, M., & Gencyilmaz, G. (2007). A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. European journal of operational research, 177(3), 1930-1947.

Thenarasu, M., Rameshkumar, K., & Anbuudayasankar, S. P. (2019). Multi-criteria decision-making approach for minimizing makespan in a large-scale press-shop. International Journal of Industrial Engineering, 26(6), 962-985.

Thenarasu, M., Rameshkumar, K., Anbuudayasankar, S. P., Arjunbarath, G., & Ashok, P. (2020). Development and selection of hybrid dispatching rule for dynamic job shop scheduling using multi-criteria decision making analysis (MCDMA). International Journal for Quality Research, 14(2), 487–504.

Thenarasu, M., Rameshkumar, K., Rousseau, J., & Anbuudayasankar, S. P. (2022). Development and analysis of priority decision rules using MCDM approach for a flexible job shop scheduling: A simulation study. Simulation Modelling Practice and Theory, 114, 102416.

Uniyal, N., Pant, S., & Kumar, A. (2020). An overview of few nature inspired optimization techniques and its reliability applications. International Journal of Mathematical, Engineering and Management Sciences, 5(4), 732-743.

Uniyal, N., Pant, S., Kumar, A., & Pant, P. (2022). Nature-inspired metaheuristic algorithms for optimization. In Kumar, A., Pant, S., Ram, M. & Yadav, O. (Eds). Meta-heuristic Optimization Techniques: Applications in Engineering, Berlin, Boston: De Gruyter, 1-10.

Vinod, V., & Sridharan, R. (2011). Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. International Journal of Production Economics, 129(1), 127-146.

Wang, Z., Qi., Y., Cui., H, Zhang., J. (2019). A hybrid algorithm for order acceptance and scheduling problem in make-to-stock/make-to-order industries, Computers & Industrial Engineering, 127(46), 841–852.

Xiong, H., Shi, S., Ren, D., & Hu, J. (2022). A survey of job shop scheduling problem: The types and models. Computers & Operations Research, 142, 105731.