Mixed-Integer Linear Programming Model for the Dual-Resource Flexible Job-Shop Scheduling Problem

Vedant Agrawal
Department of Mechanical Engineering, Indian Institute of Technology Kharagpur
West Bengal 721302, India
iitkgp.vedant@gmail.com

Alejandro Vital-Soto and Jessica Olivares-Aguila
Department of Management Science, Shannon School of Business, Cape Breton University
Sydney, NS B1M 1A2, Canada
alejandro_vital@cbu.ca, jessica_olivares@cbu.ca

Abstract

Scheduling is a vital function for efficiently operating flexible job-shop systems. Traditionally, that function considers assigning jobs to machines and their sequence. However, machines need to be operated by another set of resources (i.e., workers). Due to the fact that operators are skilled workers, the available pool is limited. Hence, the interaction of machines and humans needs to be studied in an integral manner to address the scheduling problem. In this article, a novel precedence variable-based, mixed-integer linear programming model is developed for the dual-resource flexible job-shop problem. The mathematical formulation deals with the optimal assignment of machines and workers to operations and the operation sequence in both resources by minimizing the makespan. The model gives an exact solution by solving both the assignment and the sequencing problems concurrently. The model was implemented in Docplex and was run on three instances of varying sizes. The model solved the three introduced examples, including a large instance involving 20 operations with 4 workers and 4 machines, using only 1662 variables and 5217 constraints in 156.43 seconds, indicating that the proposed model is adequate. The model can be used to label training examples for machine learning-based techniques as well as help track and compare models developed using heuristics.

Keywords
Dual-Resource, Flexible Job-shop Scheduling Problem, Makespan, Mathematical Model, and Mixed-integer linear programming.

1. Introduction
Manufacturing systems have evolved to adapt to the changing requirements, and while they have experienced several industrial revolutions and moved from manual operations to automation, human-centred manufacturing systems are still required. Skilled workers are needed to operate the systems. Hence, to optimize the efficiency of manufacturing systems, the consideration of workers' capabilities is required.

Scheduling has been an integral part of every organization, which might involve the interaction of humans and machines, the dual resources, to carry out the organization's principal function (i.e., manufacturing). Each resource group (e.g., machines or workers) can interact with one or more resources in flexible environments. The schedules for every entity of the two resource groups must be feasible and optimal to carry out the process without any clashes efficiently. Therefore, the scheduling for the dual-resource problem needs to be studied.

Various iterative or search-based methods for solving the scheduling problem involve breaking down this problem into two parts. First is the optimal assignment of a machine and worker pair to the different operations. Second is the optimal sequence of the other jobs based on the optimal assignments. However, it is to be noted that these two parts need to be solved simultaneously or concurrently because the optimal assignment depends on the optimal sequence as well. Therefore, the objective of this study is to propose a novel precedence variable-based, mixed-integer linear programming (MILP) model which provides an exact solution to the dual-resource flexible job-shop scheduling problem (DR-FJSP), solving both the parts concurrently and minimizing the makespan value while using a minimum number of decision variables.

The rest of the paper is organized as follows. Section 2 studies the literature review, highlighting the important work done in this field. Section 3 incorporates the formal problem description and the proposed mathematical model. Section 4 illustrates the numerical methods and examples for testing the mathematical model. Section 5 introduces the results and discussion on the examples solved. Finally, Section 6 presents the conclusion and future work.

2. Literature Review

Job-shop scheduling (JSP) can be dated as early as the 1950s. The simpler version of the problem involved designing schedules for jobs that need to be processed on specific machines that can handle only one type of operation. Later, a flexible approach to JSP, known as flexible job-shop scheduling problem (FJSP), was introduced where individual machines can handle more operations, making the scheduling problem more flexible and complex. JSP and FJSP aim to solve for the optimal schedule by minimizing various objectives (e.g., makespan, tardiness, and earliness) and are defined as NP-Hard (Sotskov and Shakhlevich 1995).

This field has been popular since the late 90s, with numerous contributions. Traditional approaches to solving the FJSP involved designing mathematical models and looking for feasible methods to obtain their solutions quickly. According to Demir and Kürşat Işleyen (2013), different mathematical models formulated to solve FJSP can be classified depending on the type of binary variable used for sequencing the operations. They introduced three categories: sequenced positioned variable, precedence variable, and time-indexed variable. The sequenced position variable model works with the schedule of jobs on a particular machine and defines the sequencing variable such that each operation can be fit into slots in the machine schedule (Unlu and Mason 2010; Wagner 1959). Manne (1960) introduced the precedence variable model in job shop scheduling. The precedence variable helps to define constraints based on the relative precedence of different operations scheduled on the same resource, ensuring no overlap. Furthermore, Özgüven et al. (2010) formulated a mathematical model for FJSP and FJSP with process plan flexibility using the precedence variable model. The most significant disadvantage of this approach is the computation complexity that cannot handle problems involving a greater number of machines in feasible times. Later, this field evolved by incorporating various heuristic and metaheuristic approaches, which gave an approximate solution to the problem in less time (Xie et al. 2019). Recent studies have used deep reinforcement learning and graph neural networks to solve the FJSSP by assigning multiple agents, each for every machine (Zhang et al. 2023).

Despite all the advancements to solve the FJSP, the machines' schedule is often insufficient to carry out the production process. A human worker operates the machines in most scenarios, and the work efficiency depends on the operator. Therefore, DR-FJSP was introduced to incorporate the involvement of another resource (worker) along with the machines. This helped to lay out a more practical planning strategy that considers the different efficiencies related to each worker-machine pair for each operation. Also, this helped to ensure no overlap between the schedules of different workers and machines so that different workers could work on more than one machine at different times, providing flexibility while increasing the complexity. A state-of-the-art review on dual-resource-constrained systems has been listed in the literature (Dhiflaoui et al. 2018).

Incorporating an additional resource introduced new challenges, and several studies have been developed in the previous decades concerning industry management aspects. For instance, Wang et al. (2018) presented a dual-resource constrained mathematical model for small-batch multi-variety job shops, which provides an exact solution to minimize costs and maximize efficiency based on the capacity and productivity requirements for the work center. Similarly, Wirojanagud et al. (2007) analyzed the workforce decisions about hiring, firing, and cross-training to ensure minimal loss to the organization considering dual-resourced constraints. Gomes da Silva et al. (2006) developed a multiple criteria mixed integer programming model for aggregate planning, considering practical aspects like worker training and legal restrictions on downsizing.

Yildiz and Eski (2006) worked on the dual-resource-constrained assembly lines using artificial neural networks to get optimum schedules for the arrangement of workers on different workstations, considering production efficiency metrics. Lobo et al. (2013, 2014) developed heuristics to determine a lower bound to the maximum job lateness and a possible schedule that fits the determined lower bound in a dual-resource-constrained job shop setup. Huang et al.

(2014) worked on scheduling the optimization strategy for dual-resource job-shop scheduling (DR-JSP) with heterogeneous workers by using a pheromone branch genetic algorithm.

Elmaraghy et al. (1999) applied the genetic algorithm to compare various rules and heuristics in machine and worker assignments and schedules for DR-JSP and JSP. Chaudhry and Drake (2009) gave a generalized GA implementation for DR-JSP for identical parallel machines and worker assignments. The approach was easy to implement on spreadsheets and surpassed various benchmarks.

Various approaches have been developed to solve problems involving machine-worker pairs in a flexible setup. Often, this complete problem is divided into two parts. At first, a machine-worker pair is assigned to all the operations of different jobs. Later, as a second step, an optimal schedule is laid out depending on the assignments. Now, these steps are iterated until a feasible solution is obtained. Lei and Guo (2014) implemented a variable neighborhood search (VNS) algorithm for optimizing DR-FJSP worker-machine assignments. They introduced a novel method for representing a task sequencing list in the form of a quadruple string for describing machine-worker assignments of different operations. Zheng and Wang (2016) encoded a novel knowledge-guided fruit fly search algorithm for DR-FJSP. The problem was broken into two stages: the resource assignment stage and the operation sequencing stage. The algorithm incorporated a knowledge-guided search along with a smell-based search to ensure both exploration and exploitation in the search space. Gong et al. (2018) implemented a novel hybrid genetic algorithm (NHGA) that uses a three-layer chromosome encoding method to solve the DR-FJSP optimization problem involving machine and worker schedules. The algorithm's hyperparameters were determined using Taguchi’s experimental design, and various benchmarks were set for the DR-FJSP for comparison to the proposed NHGA.

Several mathematical models for solving DR-FJSP are available. Foroutan et al. (2023) developed a mixed integer mathematical model for parallel machine scheduling while incorporating the involvement of workers for set-up and transport of jobs. Meng et al. (2019) linearly modeled the DR-FJSP using time-indexed mixed-integer decision variables. Two mathematical models were developed based on idle-time and idle-energy ideas. Further, the models were solved using a VNS algorithm. The mathematical model introduced does not clarify non-overlap constraints for the workers, and the model focuses on machine constraints only. Also, the model introduced is ideal for job sequencing and does not look into the assignments and sequencing concurrently. The number of decision variables in the time-indexed models increases with the maximum duration of processing times. Moreover, it constrains the model to divide the time into fixed intervals and thus cannot incorporate fractional processing times. Vital-Soto et al. (2022) developed a precedence variable-based linear mathematical model for DR-FJSP with sequencing flexibility. The model is solved using an elitist, non-dominated sorting genetic algorithm. The model involves a large number of decision variables, and therefore, its mathematical complexity is high.

Various search-based heuristics and meta-heuristics discussed above solve DR-FJSP in two sequential steps: assignment and sequencing. However, dividing the problem into these two steps could be more efficient. The assignment and the scheduling must be done concurrently because they depend on each other. Therefore, there is a need to develop a linear mathematical model which could provide an exact solution. It is essential to lay out the mathematical model to have a deeper understanding of the problem and to pave the way for developing advanced techniques (Unlu and Mason 2010). Also, with the invention of high computational power and the emergence of quantum computing, it might be possible to get an exact solution using mathematical models alone. Hence, this study aims to provide an alternative formulation of the DR-FJSP and a benchmark for obtaining an exact optimal solution to the problem. Moreover, it reports and analyses solving problems of different sizes and complexities.

3. Mathematical Programming Model

In the DR-FJSP problem, there is a set of N numbers of jobs, indexed from 1 to N. Each job has a defined set of operations from indexed from 1 to γ_i that needs to be performed in a strict sequence one after another. Each operation is performed from one among the set of machines indexed from 1 to M, which is operated by a worker among the set of workers indexed from 1 to W. Both machines and workers are flexible to process different kinds of operations. The workers have different efficiencies associated with various operations on different machines. The processing times of each worker-machine combination for every operation are defined beforehand and are used as parameters for the problem.

The DR-FJSP problem determines the optimal resource schedules for both the worker and machines, with a single objective function to minimize makespan.

Assumptions:

- There is no precedence relationship between operations of different jobs.
- All the machines and workers are available at time zero.
- No pre-emption is allowed.
- Setup times are negligible or incorporated in different operations' processing times.
- Infeasible worker-machine pairs are assigned with a high processing time.

Indices and parameters:

i : $Job\ Index\ \in [1, N], where\ N\ is\ the\ total\ number\ of\ jobs$

j : $Operation\ Index\ \in [1, \gamma_i], where\ \gamma_i\ is\ the\ total\ number\ of\ operations\ for\ job\ i$

u: $Worker\ Index\ \in [1, W]\ where\ W\ is\ the\ total\ number\ of\ workers$

k : $Machine\ Index\ \in [1, M]\ where\ M\ is\ the\ total\ number\ of\ machines$

O_{ij}: $operation\ j\ of\ job\ i$

t_{ijku} : $processing\ time\ of\ O_{ij}\ on\ machine\ k\ with\ worker\ u$

Decision Variables:

$$X_{ijku} = \begin{cases} 1\ if\ O_{ij}\ is\ assigned\ to\ machine\ k\ and\ worker\ u \\ 0\ otherwise \end{cases}$$

$$Y_{ijIJ-k} = \begin{cases} 1\ \ If\ O_{ij}\ is\ scheduled\ before\ O_{IJ}\ on\ machine\ k \\ 0\ otherwise \end{cases}$$

$$Z_{ijIJ-u} = \begin{cases} 1\ \ If\ O_{ij}\ is\ scheduled\ before\ O_{IJ}\ on\ worker\ u \\ 0\ otherwise \end{cases}$$

S_{ij} : $Starting\ time\ of\ O_{ij}$

C_{ij} : $Completion\ time\ of\ O_{ij}$

C_{max} : $Makespan$

Objective:

$$\text{Minimize } C_{\max} \tag{1}$$

Constraints:

$$\sum_{u=1}^{W}\sum_{k=1}^{M} X_{ijku} = 1 \quad \forall\, i \,\epsilon\, [1, \mathrm{N}], j \,\epsilon\, [1, \gamma_i] \tag{2}$$

$$S_{ij+1} \geq C_{ij}\ \forall\, i \,\epsilon\, [1, \mathrm{N}],\ j \,\epsilon\, [1, \gamma_{i-1}] \tag{3}$$

$$\sum_{u} X_{ijku} \geq Y_{ijIJ-k}\ \forall\, i \,\epsilon\, [1, \mathrm{N}-1],\ j \,\epsilon\, [1, \gamma_i], I \,\epsilon\, [i+1, N],\ \mathrm{J} \,\epsilon\, [1, \gamma_I],\ k \,\epsilon\, [1, M] \tag{4}$$

$$\sum_{u} X_{IJku} \geq Y_{ijIJ-k}\ \forall\, i \,\epsilon\, [1, \mathrm{N}-1],\ j \,\epsilon\, [1, \gamma_i], I \,\epsilon\, [i+1, N],\ \mathrm{J} \,\epsilon\, [1, \gamma_I],\ k \,\epsilon\, [1, M] \tag{5}$$

$$S_{ij} \geq C_{IJ} + \left(2 + Y_{ijIJ-k} - \left(\sum_{u} X_{ijku}\right) - \left(\sum_{u} X_{IJku}\right)\right) * L \\ \forall\, i \,\epsilon\, [1, \mathrm{N}-1],\ j \,\epsilon\, [1, \gamma_i], I \,\epsilon\, [i+1, N],\ \mathrm{J} \,\epsilon\, [1, \gamma_I],\ k \,\epsilon\, [1, M] \tag{6}$$

$$S_{IJ} \geq C_{ij} + \left(1 - Y_{ijIJ-k}\right) * L\ \forall\, i \,\epsilon\, [1, \mathrm{N}-1],\ j \,\epsilon\, [1, \gamma_i], I \,\epsilon\, [i+1, N],\ \mathrm{J} \,\epsilon\, [1, \gamma_I],\ k \,\epsilon\, [1, M] \tag{7}$$

$$\sum_{k} X_{ijku} \geq Z_{ijIJ-u}\ \forall\, i \,\epsilon\, [1, \mathrm{N}-1],\ j \,\epsilon\, [1, \gamma_i], I \,\epsilon\, [i+1, N],\ \mathrm{J} \,\epsilon\, [1, \gamma_I],\ u \,\epsilon\, [1, W] \tag{8}$$

$$\sum_k X_{IJku} \geq Z_{ijIJ-u} \; \forall \, i \in [1, \mathrm{N}-1],\; j \in [1, \gamma_i], I \in [i+1, N],\; \mathrm{J} \in [1, \gamma_I],\; u \in [1, W] \quad (9)$$

$$S_{ij} \geq C_{IJ} + \left(2 + Z_{ijIJ-u} - \left(\sum_k X_{ijku}\right) - \left(\sum_k X_{IJku}\right)\right) * L$$
$$\forall \, i \in [1, \mathrm{N}-1],\; j \in [1, \gamma_i], I \in [i+1, N],\; \mathrm{J} \in [1, \gamma_I],\; u \in [1, W] \quad (10)$$

$$S_{IJ} \geq C_{ij} + \left(1 - Z_{ijIJ-u}\right) * L \; \forall \, i \in [1, \mathrm{N}-1],\; j \in [1, \gamma_i], I \in [i+1, N],\; \mathrm{J} \in [1, \gamma_I],\; u \in [1, W] \quad (11)$$

$$C_{ij} \geq S_{ij} + \left(\sum_u \sum_k t_{ijku} * X_{ijku}\right) \forall \, i \in [1, \mathrm{N}],\; j \in [1, \gamma_i] \quad (12)$$

$$C_{max} \geq C_{ij} \; \forall \, i \in [1, \mathrm{N}],\; j \in [1, \gamma_i] \quad (13)$$

$$S_{ij} \geq 0 \; \forall \, i \in [1, \mathrm{N}],\; j \in [1, \gamma_i] \quad (14)$$

$$C_{ij} \geq 0 \; \forall \, i \in [1, \mathrm{N}],\; j \in [1, \gamma_i] \quad (15)$$

$$\mathrm{X_{ijku}} \in \{0,1\} \; \forall \, i \in [1, \mathrm{N}],\; j \in [1, \gamma_i], \mathrm{J} \in [1, \gamma_I],\; k \in [1, M], u \in [1, W] \quad (16)$$

$$Y_{\mathrm{ijIJ-k}} \in \{0,1\} \; \forall \, i \in [1, \mathrm{N}-1],\; j \in [1, \gamma_i], I \in [i+1, N],\; \mathrm{J} \in [1, \gamma_I],\; k \in [1, M] \quad (17)$$

$$Z_{\mathrm{ijIJ-u}} \in \{0,1\} \; \forall \, i \in [1, \mathrm{N}-1],\; j \in [1, \gamma_i], I \in [i+1, N],\; \mathrm{J} \in [1, \gamma_I],\; u \in [1, W] \quad (18)$$

Equation (1) presents the objective function of the MILP as the minimization of makespan. Inequality (2) ensures that only a single worker and machine pair are assigned to complete a single operation. Constraint (3) provides the linear precedence constraints for successive operations of a particular job. Constraints (4), (5), (8), and (9) are logical constraints that allow the resource precedence variables to hold unity only when the two concerned operations (O_{ij} and O_{IJ}) are scheduled on the same resource. Constraints (6) and (7) prevent overlapping between operations in the machine k. Constraints (10) and (11) ensure no operations overlap in worker u. Constraint (12) defines the completion time of an operation. Inequality (13) illustrates the makespan. Constraints (14) to (18) represent the variable types.

4. Numerical Analysis

The mathematical model was implemented and tested on three examples of varying sizes. Table 1 shows the number of jobs, operations, machines, and workers per example.

For each instance, the processing times (t_{ijku}) for each operation (O_{ij}) of job i on machine k with worker u are presented in Tables 2 to 4. In Table 2, the symbol "-" represents an infeasible assignment of the machine-worker pair, and the numerical values represent the processing time (t_{ijku}) in time units. For instance, in Table 2, for job $i = 2$, operation $j = 1$, machine k=2, and worker $u = 2$, the processing time is $t_{2122} = 7$. Moreover, Examples 2 and 3 as presented in Table 3 and Table 4, illustrate fully flexible systems as all the machines and workers can perform all the operations.

Table 1. Size complexity of the different examples

Example Number	**Number of Jobs**	**Total Number of Operations**	**Number of Machines**	**Number of Workers**
1	4	8	2	2
2	4	16	2	3
3	5	20	4	4

Table 2. Processing times for example 1

		k =1		k =2	
Jobs	O_{ij}	u =1	u =2	u =1	u =2
i =1	**O11**	-	4	2	-
	O12	-	2	-	-
i =2	**O21**	-	4	3	7
	O22	-	5	1	4
i =3	**O31**	-	-	-	2
	O32	-	-	-	4
	O33	-	-	-	5
i =4	**O41**	-	2	-	6

Table 3. Processing times for example 2

		k =1		k =2		k =3	
Jobs	O_{ij}	u =1	u =2	u =1	u =2	u =1	u =2
i =1	**O11**	2	4	1	2	4	5
	O12	5	5	2	2	4	2
	O13	5	1	5	3	2	2
	O14	1	4	1	5	4	5
i =2	**O21**	1	3	5	4	1	3
	O22	2	1	3	5	1	2
	O23	1	2	2	1	3	5
	O24	5	5	1	2	3	5
i =3	**O31**	4	2	5	2	5	3
	O32	2	3	2	2	2	1
	O33	2	5	5	2	4	5
	O34	2	2	1	4	2	4
i =4	**O41**	5	5	5	1	4	4
	O42	5	1	2	2	2	5
	O43	1	4	3	3	4	1
	O44	4	5	3	4	1	5

Example 1 was taken from (Zheng and Wang 2016). The other two examples (i.e., 2 and 3) were generated using a Python program that takes the problem sizes as input and randomly generates the processing times (t_{ijku}) using a uniform distribution. Since the processing times for each operation are industry-specific, it can have any value, depending upon the customization of the job. For more complex industries with a more flexible manufacturing setup, the processing times can hold a variety of values. Further, we generated the problem sets with all the workers being eligible to carry out each and every operation, such that there can be a large pool of feasible solutions from which we have to find the optimal one. Therefore, the generated example sets, as shown in Tables 3 and 4, are designed to be complex enough to be a good test for our proposed mathematical model.

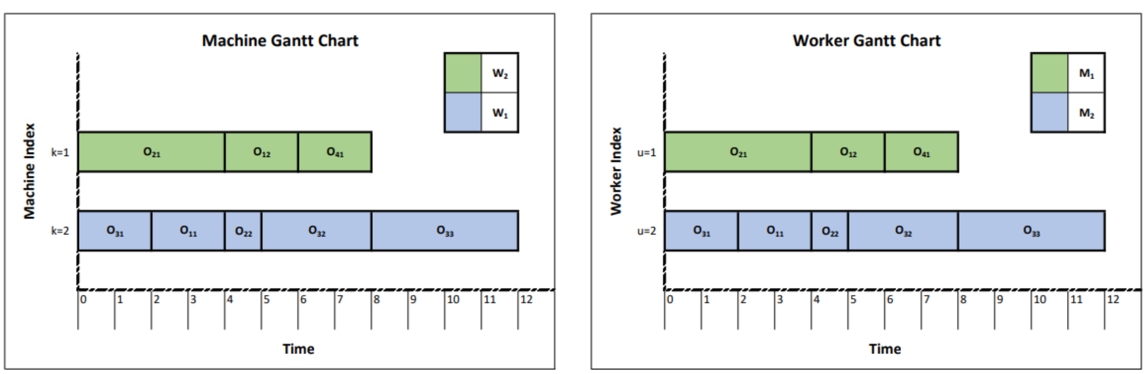
Table 4. Processing times for example 3

		k =1				k =2				k =3				k =4			
Job	O_{ij}	u =1	u =2	u =3	u =4	u =1	u =2	u =3	u =4	u =1	u =2	u =3	u =4	u =1	u =2	u =3	u =4
i =1	O_{11}	4	5	1	5	1	5	1	1	5	4	3	1	2	4	2	4
	O_{12}	1	3	4	5	3	1	2	3	3	1	2	4	4	5	2	3
	O_{13}	5	1	5	3	2	4	4	4	4	4	3	5	3	2	5	1
	O_{14}	2	5	2	4	4	1	1	4	3	5	5	2	1	2	3	4
i =2	O_{21}	1	5	2	3	1	5	2	5	1	3	5	3	3	2	5	1
	O_{22}	1	4	4	2	1	3	2	5	5	4	1	5	3	2	1	2
	O_{23}	5	1	1	1	3	4	3	3	5	4	4	5	5	4	3	4
	O_{24}	1	3	4	3	2	1	1	1	5	3	4	1	2	5	4	2
i =3	O_{31}	4	3	5	1	1	1	2	4	4	4	4	2	3	5	5	2
	O_{32}	3	4	5	1	3	1	1	4	5	4	4	2	4	3	2	5
	O_{33}	5	5	5	2	3	4	1	2	5	2	3	1	1	3	4	5
	O_{34}	4	2	4	4	5	5	1	5	1	4	3	3	5	4	2	1
i =4	O_{41}	5	3	2	2	5	4	4	5	3	4	5	3	5	2	2	3
	O_{42}	4	2	2	1	2	5	5	3	1	2	2	2	4	2	1	3
	O_{43}	5	5	2	4	4	1	1	5	2	4	5	2	2	5	4	3
	O_{44}	5	4	3	1	5	3	1	4	1	1	2	3	4	3	2	4
i =5	O_{51}	2	1	2	4	4	5	1	5	3	4	2	3	2	2	1	3
	O_{52}	5	5	4	5	2	4	5	5	5	1	4	1	5	1	2	4
	O_{53}	2	2	4	5	4	3	3	5	1	4	1	2	4	4	1	5
	O_{54}	3	3	2	4	5	5	1	3	5	3	3	3	1	5	2	4

5. Results and Discussion

The three instances mentioned above were solved using the API in DOcplex Python library provided by IBM. The Python instance was run on a PC with an Intel(R) Core(TM) i7-1065G7 CPU 1.30GHz and 8 GB of RAM. The problem was formulated using the model class in docplex.mp.model, a MILP model in the library.

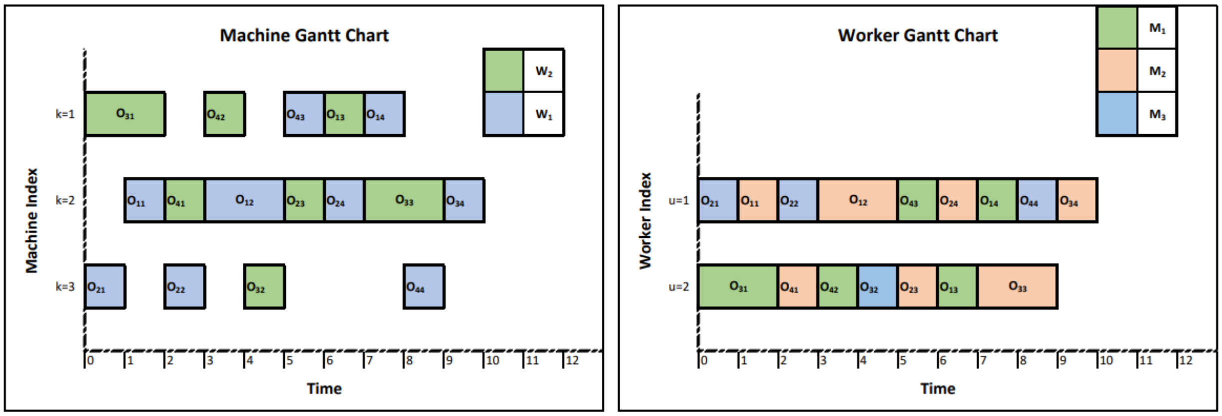
Figures 1, 2, and 3 depict the worker and machine Gantt charts solutions for each example, respectively. It can be observed that there is no overlap in the jobs in any of the charts, and the precedence constraints are satisfied. Moreover, there is minimal idle time in the resources, which increases the productivity of the workspace.



(a) (b)

Figure 1. Machine (a) and worker (b) Gantt charts for example 1

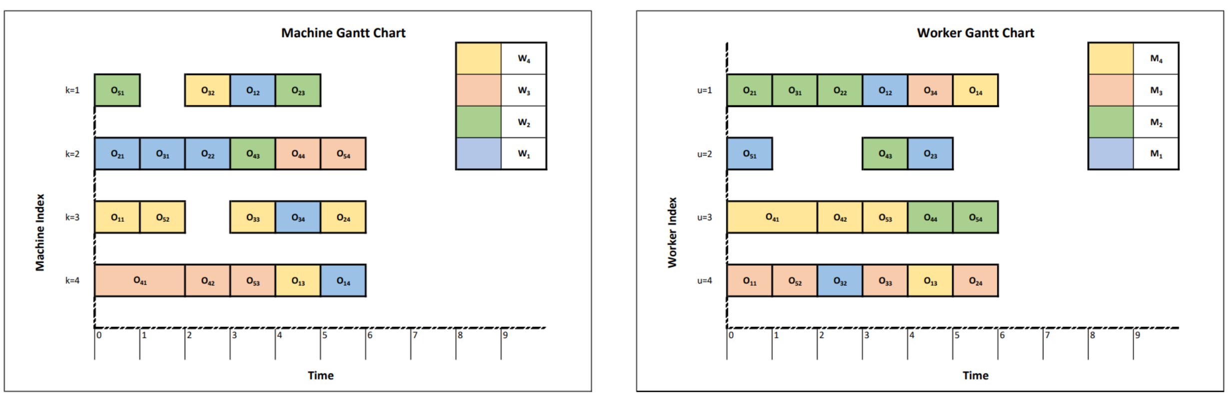
For example 2, as shown in Figure 2, it can be noted that the workers do not have idle time. In contrast, the machines have some unused time. This situation happens as the number of workers is less than the number of machines.



(a) (b)

Figure 2. Machine (a) and worker Gantt (b) charts for example 2

Also, since examples 2 and 3 are generated using a uniform distribution, increasing the number of worker-machine combinations also increases the possibility of having a machine-worker pair having a lower processing time, which can fit into an optimal schedule. Therefore, it can be seen in example 3 Gantt chart that almost all the selected machine-worker pairs have processing times equal to one, which is the lowest possible processing time value. Hence, the total makespan is also the lowest for the third example, even after having a greater number of operations.



(a) (b)

Figure 3. Machine (a) and worker (b) Gantt charts for example 3

The number of variables, constraints, optimal makespan value, and time taken to solve the examples are given in Table 5. It can be seen that the number of variables and constraints increases with an increase in operations, workers, and machines. Moreover, Table 5 shows the computational time in seconds (s).

Table 5. Summary table of solutions

Example	No. of Variables	No. of Constraints	Makespan	Time Taken(s)
1	150	406	12	0.4
2	626	1998	10	94.9
3	1662	5217	6	156.43

Table 5 provides insights into the algorithm's performance and its ability to solve scheduling instances of varying complexity by using the minimum number of variables and constraints. These results are essential for evaluating the algorithm's efficiency and effectiveness in real-world scenarios. Furthermore, they serve as a basis for comparing the proposed approach to existing methods and showcasing the potential advantages of the proposed algorithm.

6. Conclusion

This study provides a novel MILP that offers an exact solution for DR-FJSP. The model is formulated using mixed integer and precedence variables and optimizes the total makespan. The model succeeded in providing optimal machine - worker assignments and schedules. Moreover, the model was tested on problems of various sizes, and their solutions were reported in the form of both machine and worker Gantt charts, along with their solving times. It is shown that this model uses a minimum number of variables to solve this problem, making it both simpler and faster.

It is worth mentioning that the model does not use worker-machine eligibility sets; instead, a high processing time value is assigned for infeasible pairs. This can cause a problem if the size of the problem grows and the total makespan is comparable to the high value assigned for the infeasible pair. Hence, measures to deal with this problem need to be devised.

Future research includes the development of efficient solution methods, such as metaheuristics, hyperheuristics, and the integration with machine learning. For instance, this model will provide optimal solutions for labeling training examples required in supervised machine learning models. Also, the model presented is uniform and unbiased towards the two kinds of resources. Therefore, this can be extended to formulate a general model for a multi-resource flexible job-shop scheduling problem.

Acknowledgments

The authors want to acknowledge the Mitacs Globalink program for providing funding to the first author.

References

Chaudhry, I. A. and Drake, P. R., Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms, *International Journal of Advanced Manufacturing Technology*, vol. 42, nos. 5–6, pp. 581–594, 2009.

Demir, Y. and Kürşat Işleyen, S.,Evaluation of mathematical models for flexible job-shop scheduling problems, *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 977–988, 2013.

Dhiflaoui, M., Nouri, H. E., and Driss, O. B., Dual-Resource Constraints in Classical and Flexible Job Shop Problems: A State-of-the-Art Review., *Procedia Computer Science*, vol. 126, pp. 1507-1515, 2018.

Elmaraghy, H., Patel, V., and Abdallah, B. 1999. A Genetic Algorithm Based Approach for Scheduling of Dual-Resource Constrainded Manufacturing Systems, *Annals of the CIRP,* vol. 48, no. 1, 1999.

Foroutan, R. A., Rezaeian, J., and Shafipour, M., Bi-objective unrelated parallel machines scheduling problem with worker allocation and sequence dependent setup times considering machine eligibility and precedence constraints, *Journal of Industrial and Management Optimization*, vol. 19, no. 1, pp. 402–436, 2023.

Gomes da Silva, C., Figueira, J., Lisboa, J., and Barman, S., An interactive decision support system for an aggregate production planning model based on multiple criteria mixed integer linear programming, *Omega*, vol. 34, no. 2, pp. 167–177, 2006.

Gong, G., Deng, Q., Gong, X., Liu, W., and Ren, Q., A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators, *Journal of Cleaner Production*, vol. 174, pp. 560–576, 2018.

Huang, Y., Yan, X. T., and Li, J. Y. 2014. Dual Resource Constrained Job Shop Scheduling Based on Compressed Time Window, *Advanced Materials Research*, vol. 905, pp. 731–735, 2014.

Lei, D. and Guo, X., Variable neighbourhood search for dual-resource constrained flexible job shop scheduling, *International Journal of Production Research*, vol. 52, no. 9, pp. 2519–2529, 2014.

Lobo, B. J., Hodgson, T. J., King, R. E., Thoney, K. A., and Wilson, J. R., An effective lower bound on Lmax in a worker-constrained job shop, *Computers and Operations Research*, vol. 40, no. 1, pp. 328–343, 2013.

Lobo, B. J., Wilson, J. R., Thoney, K. A., Hodgson, T. J., and King, R. E., A practical method for evaluating worker allocations in large-scale dual resource constrained job shops, *IIE Transactions*, vol. 46, no. 11, pp. 1209–1226, 2014.

Manne, A. S., On the Job-Shop Scheduling Problem, *Operations Research*, vol. 8, no. 2, pp. 219–223, 1960.

Meng, L., Zhang, C., Zhang, B., and Ren, Y., Mathematical Modeling and Optimization of Energy-Conscious Flexible Job Shop Scheduling Problem with Worker Flexibility, *IEEE Access*, vol. 7, pp. 68043–68059, 2019.

Özgüven, C., Özbakir, L., and Yavuz, Y., Mathematical models for job-shop scheduling problems with routing and process plan flexibility, *Applied Mathematical Modelling*, vol. 34, no. 6, pp. 1539–1548, 2010.

Sotskov, Y. N. and Shakhlevich, N. V., NP-hardness of shop-scheduling problems with three jobs, *Discrete Applied Mathematics*, vol. 59, pp. 237-266, 1995.

Unlu, Y. and Mason, S. J., Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems, *Computers and Industrial Engineering*, vol. 58, no. 4, pp. 785–800, 2010.

Vital-Soto, A., Baki, M. F., and Azab, A., A multi-objective mathematical model and evolutionary algorithm for the dual-resource flexible job-shop scheduling problem with sequencing flexibility, *Flexible Services and Manufacturing Journal*, 2022.

Wagner, H. M., An integer linear-programming model for machine scheduling, *Naval Research Logistics Quarterly,* vol. 6, no. 2, pp. 131-140, 1959.

Wang, Y., Cen, H. J., and Yang, O., Optimal configuration for workshop manufacturing system under dual resource constraints, *International Journal of Simulation Modelling*, vol. 17, no. 1, pp. 180–189, 2018.

Wirojanagud, P., Gel, E. S., Fowler, J. W., and Cardy, R., Modelling inherent worker differences for workforce planning, *International Journal of Production Research*, vol. 45, no. 3, pp. 525–553, 2007

Xie, J., Gao, L., Peng, K., Li, X., and Li, H., Review on flexible job shop scheduling, *IET Collaborative Intelligent Manufacturing*, vol. 1, no. 3, pp. 67–77, 2019.

Yildiz Gokalp and Eski, O., An Artificial Neural Network Based Simulation Metamodeling Approach for Dual Resource Constrained Assembly Line, pp. 1002–11, in Kollias Stefanos and Stafylopatis, A. and D. W. and O. E. (ed.), *Artificial Neural Networks – ICANN 2006*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2006.

Zhang, J. D., He, Z., Chan, W. H., and Chow, C. Y., DeepMAG: Deep reinforcement learning with multi-agent graphs for flexible job shop scheduling, *Knowledge-Based Systems*, vol. 259, 2023.

Zheng, X. L. and Wang, L., A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem, *International Journal of Production Research*, vol. 54, no. 18, pp. 5554–5566, 2016.

Biographies

Vedant Agrawal is currently a student in the Department of Mechanical Engineering at Indian Institute of Technology Kharagpur. He is pursuing his undergraduate studies in the B.Tech course of Manufacturing Science and Engineering. His research interest lies in the fourth industrial revolution, and he wants to contribute towards industrial automation. He is curious about artificial intelligence and machine learning and is working towards the integration of these new technologies in the MSME and SME industries.

Alejandro Vital Soto received a Ph.D. in industrial and manufacturing systems engineering from the University of Windsor, Ontario, Canada, in 2019. Since 2019, he has worked as an Assistant Professor of Operations Research and Management Science at the Shannon School of Business at Cape Breton University. His research interests include manufacturing systems design, optimization, scheduling, and Industry 4.0.

Jessica Olivares-Aguila received a Ph.D. in industrial and manufacturing systems engineering from the University of Windsor, Ontario, Canada, in 2018. She has worked as an Assistant Professor of Supply Chain Management at Cape Breton University since 2019. Her main research interests include supply chain design, supply chain risk management, disruption risks, product design, digital twins, and Industry 4.0.