MILP Models for Flexible Job Shop Scheduling with Spatial Constraints and Sequence Flexibility*

Yunjun Han[2], Shaoming Peng[1,2,+], Zhen Shen[2], Zhikun Tao[1,2] and Gang Xiong[2,⋆]

Abstract— Within the evolving landscape of Industry 4.0, the significance of flexible job shop scheduling problems is on the rise. This study addresses a novel category of NP-hard scheduling problems: the flexible job shop scheduling problems that incorporate site routing and operation sequencing flexibility. These problems are crucial and applicable in various contexts, including carrier-based aircraft scheduling, shipbuilding assembly scheduling, and cutting tool operation scheduling. For these problems, we present two new MILP models: Model-1 is a time-indexed model, and Model-2 is a precedence variable-based model. These models are evaluated under ten cases of the problems. Our findings indicate that the two models exhibit varying solution efficiencies when dealing with problems of diverse magnitudes. Model-2 performs better than Model-1 on small-scale problems, while Model-1 outperforms Model-2 on medium-scale problems.

I. INTRODUCTION

Flexible Job Shop Scheduling is recognized as an efficient strategy to tackle the challenges of mass customization manufacturing in the era of Industry 4.0 [1], underscoring the importance of addressing Flexible Job Shop Scheduling Problems (FJSPs).

In many flexible job shops, jobs require not only machines but also certain sites to complete operations (Fig. 1a). The transportation times of a job to the next position cannot be ignored. These problems not only include machine scheduling but also site routing (for instance in Fig. 1b, when the job is assigned to machine M3 for processing, only S2 and S3, two available sites, can be utilized for placement.). Moreover, the order of operations for a job is not single and fixed (Fig. 2(a)); on the contrary, it can generate multiple sequencing options (Fig. 2) based on the dependencies (maybe also conflict relationships) described by a DAG (Fig. 2(b)). For each job, its site routing and operation sequencing are new assumptions in the FJSPs discussed in this paper. For many flexible processing workshops such as airport aircraft operations, shipyard operations, and cutting tool workshops, this assumption is an important and practical issue (for example, see [2], [3]).

*This work was supported by the National Key Research and Development Program of China (No. 2021YFB3301504); Beijing Natural Science Foundation under Grant L233005; CAS STS Dongguan Joint Project under Grant 2021160020002

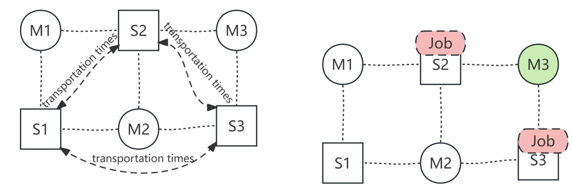
[1]Shaoming Peng and Zhikun Tao are with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China pengshaoming2021@ia.ac.cn

[2]All authors with State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China yunjun.han@ia.ac.cn

[+] Shaoming Peng is the co-first author

⋆ Gang Xiong is the corresponding author gang.xiong@ia.ac.cn

This paper refers to the above types of problems as the FJSP with spatial constraints and sequence flexibility (FJSP-SSF).



(a) network between sites and machines (b) site routing on a machine

Fig. 1: Site routing on a machine for a job

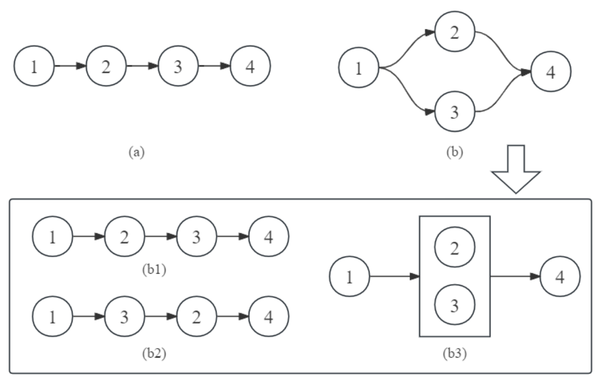


Fig. 2: Multiple operation sequences from DAG

FJSP-SSF can be addressed through two primary approaches: exact methods and approximation methods. Integer Linear Programming (ILP) models, which represent exact methods, are rendered ineffective for solving large-scale problems due to their substantial computational demands and memory requirements, while approximation methods can obtain a good near-optimal solution within a shorter time frame.

Despite their limitations in addressing large-scale issues, the study of Mixed ILP (MILP) models holds significant value. The rationale behind this is the ability of MILP models to produce optimal solutions for smaller problems, which makes them a benchmark for evaluating the effectiveness of approximation approaches. Moreover, MILP models form the basis for understanding scheduling challenges, offering a concise representation of all the intricacies within these problems and as a pivotal tool for probing and unraveling the underlying dispatch rules that govern these scheduling scenarios. With the rapid advancement of computers and artificial intelligence technologies, the application of mathematical programming and learning algorithms to solve scheduling issues has increasingly caught the attention of researchers.

[4] categorized ILP models into three types based on binary variable definitions: sequence-position variable-based model (SP), precedence variable-based model (PV), and time-indexed model (TI). The SP model's sequencing variables are defined based on the operation slots on machines. The PV model specifically defines a set of precedence variables to handle the sequencing of operations on the same machine. The TI model determines the operation sequence on machines by incorporating a time index.

For the SP model, Fattahi et al. employed this modeling technique to describe FJSP [5] and FJSP problems with overlapping [6]. According to [4], the performance of SP modeling tests is worse than that of the PV model, and there is less literature on SP than on PV. Regarding the PV model, [7] studied two NP-hard scheduling problems: the FJSPs with routing and sequencing sub-problems and the flexible job shop problem with process planning selection sub-problems, developing a model based on precedence variables. [8] improved upon Ozguven's model by decoupling the precedence variables from machines, thus reducing the model's computation time and improving solution quality. For the TI model, which is based on time indices, a typical feature is that the decision variables in the model include periods. [9] and [10] constructed ILP models for FJSPs based on time index variables. [4] tested the three models on randomly generated test instances by [5], showing that the model based on precedence variables had the shortest computation time, with the model developed by Ozgüven et al. performing optimally.

From the perspective of the research problem nature, [11] discussed and modeled the total energy consumption of flexible job shops, proposing six new MILP models from the perspectives of idle time and energy variables. Experiments showed that modeling based on idle energy performed better. [12] studied FJSP with procedural order flexibility, considering the priority between operations in terms of directed acyclic graphs rather than sequences, and proposed a MILP model to minimize the weighted delay. A bio-inspired hybrid bacterial foraging optimization algorithm was used to optimize this model. [13], [14], and [15] considered FJSPs with transport vehicle and robot path optimization, each proposing ILP models and demonstrating their efficiency in solving small-scale instances.

In terms of solution method research, [16] proposed a novel ILP model optimization method to minimize the total weighted delay. The new method has fewer decision variables and constraints. [17] established a new method for tightening model constraints during data preprocessing to improve solution performance. [18] integrated deep neural networks into the decomposition and coordination framework as an alternative to Lagrangian relaxation to better predict sub-problem solutions.

In summary, the existing models can only describe either FJSP with transportation path planning or FJSP with process flexibility, but they cannot formulate the characteristics of FJSP-SSF.

In this paper, we improve the model developed by [8], enabling it to solve FJSP-SSF, and the resulting model is referred to as Model-2. Additionally, we have also developed a time-indexed ILP model, named Model-1. We designed ten instances to test these two models. The main contributions of this paper are as follows:

- We propose a linearization modeling method for the FJSP with both spatial constraints and sequence flexibility and present two efficient MILP models. Both of these MILP models are based on two modeling ideas and incorporate the latest modeling techniques, with the modeling process of each MILP model being described in detail.
- Comparisons are made between all the proposed MILP models in terms of model size and computational complexities, and the performance of the models under different settings is analyzed.

II. PROBLEM DESCRIPTION

FJSP-SSF consists of a set of $|M|$ machines $M = \{M_k\}_{k=1}^{|M|}$ and $|S|$ processing sites $S = \{S_n\}_{n=1}^{|S|}$, each machine (e.g. machine k) having a subset of sites $S_k \in S$ available for placing jobs to perform their operations. The transportation times $\tau_{nn'}$ of the job must be considered when moving a job from site n to site n'. There is a total of $|J|$ independent jobs $J = \{J_i\}_{i=1}^{|J|}$ that need to be processed. job i requires several $|O_i|$ operations $O_i = \{O_{ij}\}_{j=1}^{|O_i|}$ to be performed, whose precedence can be described by a DAG (Directed Acyclic Graph, see Fig. 2(b)). The relationship between operations, rather than a predefined order in FJSP, may be parallel, incompatible, or sequential, leading to multiple alternative operational sequences, that is, the order of operations is flexible. Operation j of job i (O_{ij}) can be performed on any of the machines $M_{ij} \in M$ for a given processing time p_{ijk}. The notations formulating the problem are given in Table I.

The following assumptions are made in FJSP-SSF: One machine can process only one operation at a time. Each operation cannot be interrupted during work. Each site can only be placed one job at a time.

FJSP-SSF includes four sub-problems: machine selection, site routing, operation sequencing, and the sequencing of operation times based on machines, and the objective is to minimize C_{max}. It is equivalent to joint problems of multiple FJSPs (NP-hard problem [19]), making it an NP-hard problem.

TABLE I: Notation

Notation	Description
	Indices and sets
i, i'	index of jobs.
j, j'	index of operations.
k, k'	index of machines.
n, n'	index of processing sites.
t, u	index of period.
J	the set of jobs, where $i \in J$.
O_i	the set of operations of job i.
M	the set of machines, where $k \in M$.
S	the set of processing sites, where $n \in S$.
M_{ij}	the set of machines on which operation j of job i can be processed $M_{ij} \subseteq M$.
S_k	the set of sites where the machine k can be placed, $S_k \subseteq S$.
p_{ijk}	the processing time of operation j of job i on machine k.
D_i	the set of predecessor pair of job i, $(j, j') \in D_i$ indicates that operation j precedes operation j'.
N_i	the set of conflict operation pair of job i, $(j, j') \in D_i$ denotes that operation j conflicts operation j'.
$\tau_{nn'}$	the transportation time from site n to site n'.
C_{max}	maximum completion time over all jobs (makespan).
δ	time interval in one period.
T	number of period.
M	a large positive number.
	Decision variables
x_{ijknt}	binary variable; 1, if operation j of job i is processed on machine k and site n at the starting period t; 0 otherwise.
v_{ijkn}	binary variable; 1, if operation j of job i is processed on machine k and site n; 0 otherwise.
$y_{iji'j'}$	binary variable; it equals 1 if operation (i, j) precedes operation (i', j'), and 0 otherwise.
c_{ijkn}	continuous variable; the completion time of operation j of job i on machine k and site n.

III. MATHEMATICAL FORMULATIONS

We present two MILP models for solving FJSP-SSFs, one is a time-indexed model, called Model-1, and another employs precedence variables, referred to as Model-2.

A. Model-1

Model-1 embeds index of time period t into decision variables x_{ijknt}, and the model is designed as follows:

$$x_{ijknt} \in \{0,1\}, \forall i \in J, \forall j \in O_i, \forall k \in M_{ij}, \forall n \in S_k, \forall t \in T. \quad (1)$$

Constraint (1) defines the binary variables x_{ijknt}.

$$C_{max} \geq \sum_{k \in M_{ij}} \sum_{n \in S_k} \sum_{t \in T} x_{ijknt} \cdot (t + p_{ijk}), \forall i \in J, \forall j \in O_i. \quad (2)$$

Constraint (2) determines the makespan.

$$\sum_{k \in M_{ij}} \sum_{n \in S_k} \sum_{t \in T} x_{ijknt} = 1, \forall i \in J, \forall j \in O_i. \quad (3)$$

Constraint (3) indicates that all operations must be completed and executed only once.

$$x_{ijknt} + \sum_{k' \in M_{ij'}} \sum_{n' \in S_{k'}} \sum_{u=0}^{t+p_{ijk}+\tau_{nn'}} x_{ij'k'n'u} \leq 1, \quad (4)$$
$$\forall i \in J, \forall (j, j') \in D_i, \forall k \in M_{ij}, \forall n \in S_k, \forall t \in T.$$

Constraint (4) indicates that the successor operation can only start after the completion of the preceding operation.

$$x_{ijknt} + \sum_{k' \in M_{ij'}} \sum_{n' \in S_{k'}} \sum_{u=0}^{u+p_{ijk}+\tau_{nn'}} x_{ij'k'n'u} \leq 1, \quad (5)$$
$$\forall j_i \in J, \forall (j, j') \in N_i, \forall k \in M_{ij}, \forall n \in S_k, \forall t \in T.$$

Constraint (5) indicates that conflict operations cannot start within the time range of $(t, t + p_{ijk} + \tau_{nn'})$ when the operation j is being performed. Constraint (4) and Constraint (5) consider the transportation times $\tau_{nn'}$ when the job is moved from site n to site n'.

$$(1+M) \cdot x_{ijknt} - M + \sum_{i' \neq i}^{J} \sum_{j' \in O_{i'}} \sum_{k' \in M} \sum_{u=t}^{t+p_{ijk}-1} x_{i'j'nk'u} \leq 1, \quad (6)$$
$$\forall i \in J, \forall j \in O_i, \forall k \in M_{ij}, \forall n \in S_k \cap S_{k'}, \forall t \in T.$$

$$(1+M) \cdot x_{ijknt} - M + \sum_{n' \neq n}^{S} \sum_{j' \in O} \sum_{n' \in M} \sum_{u=t}^{t+p_{ijk}-1} x_{ij'n'k'u} \leq 1, \quad (7)$$
$$\forall i \in J, \forall j \in O_i, \forall k \in M_{ij}, \forall n \in S_k \cap S_{k'}, \forall t \in T.$$

Constraint (6) and constraint (7) respectively represent that at the same time, a site only serves one job, and one job only occupies one site.

$$(1+M) \cdot x_{ijknt} - M + \sum_{i' \neq i}^{J} \sum_{j' \in O'_i} \sum_{n' \in S_k} \sum_{u=t}^{t+p_{ijk}-1} x_{i'j'n'ku} \leq 1, \quad (8)$$
$$\forall i \in J, \forall j \in O_i, \forall k \in M_{ij} \cap M_{i'j'}, \forall n \in S_k, \forall t \in T.$$

Constraint (8) indicates that at the same time, one machine only serves one job.

B. Model-2

Model-2 introduces the precedence variable $y_{iji'j'}$ from the [8] to describe the sequencing of process operations. The decision variables of Model-2 include c_{ijkn}, v_{ijkn}, and $y_{iji'j'}$.

$$c_{ijkn} \geq 0, \forall i \in J, \forall j \in O_i, \forall k \in M_{ij}, \forall n \in S_k. \quad (9)$$

Constraint (9) defines a set of non-negativity variables of c_{ijkn}.

$$v_{ijkn} \in \{0,1\}, \forall i \in J, \forall j \in O_i, \forall k \in M_{ij}, \forall n \in S_k. \quad (10)$$
$$y_{iji'j'} \in \{0,1\}, \forall i \in J, \forall i' > i, \forall j \in O_i, \forall j' \in O_{i'}. \quad (11)$$

Constraint (10) and (11) define the set of binary variables of v_{ijkn} and $y_{iji'j'}$, respectively.

$$C_{max} \geq \sum_{k \in M_{ij}} \sum_{n \in S_k} \sum_{t \in T} c_{ijkn}, \forall i \in J, \forall j \in O_i. \quad (12)$$

Constraint (12) determines the makespan, that is, the makespan must be greater than any completion time of the last operation of each job.

$$\sum_{k\in M_{ij}}\sum_{n\in S_k} v_{ijkn}=1, \forall i\in J, \forall j\in O_i. \tag{13}$$

Constraint (13) means that each operation requires performing and is assigned to exactly one machine and one site.

$$c_{ijkn}-M\cdot v_{ijkn}\leq 0, \forall i\in J, \forall j\in O_i, \forall k\in M_{ij}, \forall n\in S_k. \tag{14}$$

$$M\cdot(1-v_{ijkn})+(c_{ijkn}-p_{ijk})\geq 0, \quad \forall i\in J, \forall j\in O_i, \forall k\in M_{ij}, \forall n\in S_k. \tag{15}$$

If operation (i,j) is not performed on machine k, its completion time on machine k is forced to zero by constraint (14). Constraint (15) ensures that the completion time of each operation should be greater than or equal to its processing time.

$$\sum_{k\in M_{ij}} c_{ijkn}+\tau_{nn'}\cdot\sum_{k\in M_{ij}} v_{ijkn}-(c_{ij'k'n'}-p_{ij'k'})\leq M\cdot(1-v_{ij'k'n'}), \quad \forall i\in J, \forall (j,j')\in D_i, \forall n\in S_k, \forall n'\in S_{k'}. \tag{16}$$

$$(c_{ijkn}-p_{ijk})-c_{ij'k'n'}-\tau_{n'n} \geq (-M)\cdot(1-y_{ij'ij})-(M)\cdot(2-v_{ijkn}-v_{ij'k'n'}), \quad \forall i\in J, \forall (j,j')\in N_i, \forall k\in M_{ij}, \forall n\in S_k, \forall k'\in M_{ij'}, \forall n'\in S_{k'}. \tag{17}$$

$$(c_{ij'k'n'}-p_{ij'k'})-c_{ijkn}-\tau_{nn'} \geq (-M)\cdot(y_{ij'ij})-(M)\cdot(2-v_{ijkn}-v_{ij'k'n'}), \quad \forall i\in J, \forall (j,j')\in N_i, \forall k\in M_{ij}, \forall n\in S_k, \forall k'\in M_{ij'}, \forall n'\in S_{k'}. \tag{18}$$

We generate a set of predecessor and successor operation pairs based on the operation sequence DAG diagram, and constraint (16) ensures that the actual processing order satisfies the precedence of the operations in the collection. For incompatible pairs of operations, Constraint (17) and (18) ensures that they can be processed in either order but not in parallel.

$$\sum_{k\in M_{ij}}(c_{ijkn}-p_{ijk}\cdot v_{ijkn})-\sum_{k'\in M_{i'j'}} c_{i'j'k'n} \geq (-M)\cdot(1-y_{i'j'ij})-(M)\cdot(2-v_{ijkn}-v_{i'j'k'n}), \quad \forall i<i', \forall j\in O_i, \forall j'\in O_{i'}, \forall k\in M_{ij}, \forall k'\in M_{i'j'}, \forall n\in S_k\cap S_{k'}. \tag{19}$$

$$\sum_{k'\in M_{i'j'}}(c_{i'j'k'n}-p_{i'j'k'}\cdot v_{i'j'k'n})-\sum_{k\in M_{ij}}(c_{ijkn}) \geq (-M)\cdot(y_{i'j'ij})-(M)\cdot(2-v_{ijkn}-v_{i'j'k'n}), \quad \forall i<i', \forall j\in O_i, \forall j'\in O_{i'}, \forall k\in M_{ij}, \forall k'\in M_{i'j'}, \forall n\in S_k\cap S_{k'}. \tag{20}$$

$$\sum_{k\in M_{ij}}(c_{ijkn}-p_{ijk}\cdot v_{ijkn})-\sum_{k'\in M_{ij'}} c_{ij'k'n'} \geq (-M)\cdot(1-y_{ij'ij})-(M)\cdot(2-v_{ijkn}-v_{ij'k'n'}), \quad \forall i\in J, \forall j\in O_i, \forall j'>j, \forall k\in M_{ij}, \forall k'\in M_{i'j'}, \forall n\in S_k, \forall n'\in S_{k'}. \tag{21}$$

$$\sum_{k'\in M_{i'j'}}(c_{i'j'k'n'}-p_{i'j'k'}\cdot v_{i'j'k'n'})-\sum_{k\in M_{ij}}(c_{ijkn}) \geq (-M)\cdot(y_{ij'ij})-(M)\cdot(2-v_{ijkn}-v_{i'j'k'n'}), \quad \forall i\in J, \forall j\in O_i, \forall j'>j, \forall k\in M_{ij}, \forall k'\in M_{i'j'}, \forall n\in S_k, \forall n'\in S_{k'}. \tag{22}$$

Constraints (19) and (20) ensure that a site is only occupied by one job at the same time, and constraints (21) and (22) ensure that a job only occupies one site at a time.

$$(\sum_{n'\in S_k} c_{i'j'kn'})-p_{i'j'k}-\sum_{n\in S_k}(c_{ijkn}) \geq (-M)\cdot(1-y_{iji'j'})-(M)\cdot(2-\sum_{n\in S_k} v_{ijkn}-\sum_{n'\in S_k} v_{i'j'kn'}), \quad \forall i<i', \forall j\in O_i, \forall j'\in O_{i'}, \forall k\in M_{ij}\cap M_{i'j'}. \tag{23}$$

$$(\sum_{n\in S_k} c_{ijkn})-p_{ijk}-\sum_{n'\in S_k} c_{i'j'kn'} \geq (-M)\cdot(y_{iji'j'})-(M)\cdot(2-\sum_{n\in S_k} v_{ijkn}-\sum_{n'\in S_k} v_{i'j'kn'}), \quad \forall i<i', \forall j\in O_i, \forall j'\in O_{i'}, \forall k\in M_{ij}\cap M_{i'j'}. \tag{24}$$

Constraint (23) and (24) define the sequencing of any two operations performed on one machine and guarantee that only one of them can use the machine at the same time. For each operation pair (i,j) and (i',j'), $(i\neq i')$, constraint (23) is active only if the two operations can be performed on the same machine (k) (i.e. $\sum_{n\in S_k} v_{ijkn}=\sum_{n'\in S_k} v_{i'j'kn'}=1$) and operation (i,j) precedes operation (i',j') (i.e. $(y_{iji'j'}=1)$; otherwise, this constraint is inactive and constraint (24) is active.

IV. COMPUTATIONAL RESULTS

We generate ten hypothetical FJSP-SSFs with different sizes to analyze the performance of the proposed models. Table II shows the machine routing, site routing, and sequencing flexibility levels concerning each problem size. The details of the test problem PJ3M5S5 are given in Table III. This test problem is with three jobs, twelve operations, five machines, and five sites. There are more than two alternative sequences with a given sequence set of operations for jobs 1 and 2 and two or three alternative machines for each operation. There exist three positions to site jobs for machines 1, 3, and 5. Transportation times between sites are given in the last column. The CBC MILP Solver (Coin-or Branch and Cut Solver, initiated by John Forrest of IBM Research and has been maintained and developed by the Coin-OR organization, Version: 2.10.3, Build Date: Dec 15, 2019) is used to solve these problems. The problems have been run on a laptop computer with Intel(R) Core(TM) i7-9750H CPU and 2.59 GHz processor (16 GB RAM). Running time is limited to 3600 s to compare Model-1 with Model-2 on the same grounds. For Model-1, the scheduling horizon is divided into periods $1,2,\cdots,T$, where T can be estimated by any technique and is chosen by the researcher. In this paper, we choose 15 periods as the scheduling horizon for all test problems.

The solutions obtained by Model-1 and Model-2 for PJ3M5S5 and the corresponding Gantt charts are presented in Fig. ?? and Fig. ??, respectively. Their

TABLE II: Sites and operation sequence flexibility levels

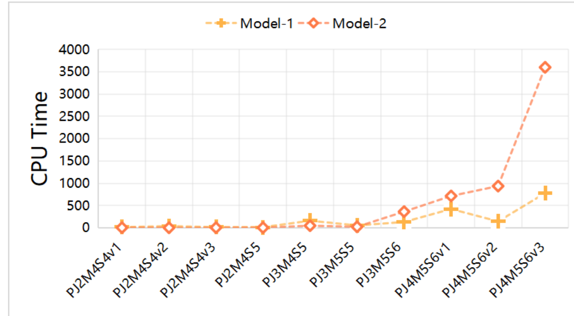
Problem No.	Jobs	Operations	Machines	Sites	Op-seqs[a]	Op-macs[b]
PJ2M4S4v1	2	8	4	4	1-2	2
PJ2M4S4v2	2	8	4	4	1-2	2
PJ2M4S4v3	2	9	4	4	1-2	2
PJ2M4S5	2	8	4	5	1-2	2
PJ3M4S5	3	12	4	5	1-3	2
PJ3M5S5	3	12	5	5	1-3	2-3
PJ3M5S6*	3	12	5	6	1-3	2-3
PJ4M5S6v1*	4	15	5	6	1-3	2-3
PJ4M5S6v2*	4	16	5	6	1-3	2-3
PJ4M5S6v3*	4	16	5	6	1-3	2-3

[a] Number of the alternative sequences for each job.
[b] Number of alternative machines for each operation.
* Medium-sized cases.

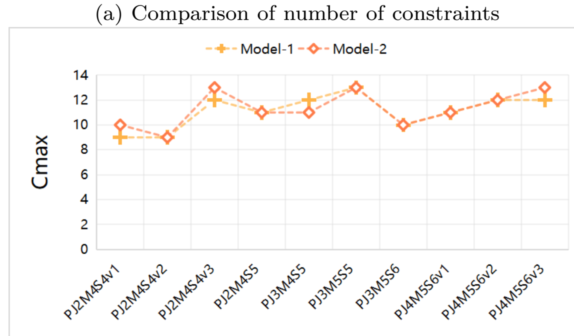
TABLE III: The data for PJ3M5S5

Sequences for jobs			Alternative machines for operations		Sites for machines		Transportation times
Jobs	No.[a]	Sequences	Operations	Machines	Machines	Sites	
1	1-1	1→2→3→4	11	1(1), 4(2)	1	1, 4, 5	1-2(1)[b],1-3(1),1-4(2), 1-5(2), 2-1(1),2-3(1), 2-4(1),2-5(2), 3-1(1), 3-2(1),3-4(1),3-5(1), 4-1(1),4-2(1),4-3(1), 4-5(1), 5-1(2),5-2(2), 5-3(1),5-4(1)
	1-2	1→3→2→4	12	3(5), 4(4), 5(4)	2	2	
			13	1(4), 2(3)	3	2, 3, 5	
			14	2(4), 3(3), 5(2)	4	4	
2	2-1	1→2→3→4	21	1(1), 4(1)	5	2, 3, 5	
	2-2	1→3→2→4	22	3(6), 4(4), 5(3)			
	2-3	1→(2‖3)→4	23	1(8), 2(6)			
			24	2(3), 3(3), 5(1)			
3	3-1	1→2→3→4	31	1(3), 4(1)			
			32	3(3), 4(5), 5(4)			
			33	1(5), 2(6)			
			34	2(2), 3(4), 5(2)			

[a] Sequence number.
[b] Outside the bracket is the machine number, and the time required to complete the operation is inside the bracket.



(a) Comparison of number of constraints



(b) Comparison of CPU time

Fig. 3: Comparison of performance for two models

resulting Cmax values are all 13, indicating that there are multiple optimal solutions to this problem. They all choose sequence No. 1-2 for job 1 and No. 2-3 for job 2.

The solutions obtained for all problems are compared in Table IV. As shown in Fig. 3b, The Cmax values obtained by Model-1 and Model-2 for all problems are overlapping, except for minor differences. However, a significant difference occurs between the two models in terms of CPU times. for small-sized problems, Model-2 significantly dominates Model-1 in terms of CPU times. However, for medium-sized problems, Model-1 consumes significantly less time than Model-2 under the same CPU time limit (3600 s). Model-1 has more constraints and variables than Model-2. Generally, this would result in a longer solution time for Model 1 than Model 2. However, for medium-scale cases, the results are the opposite. This may be because the lower proportion of non-binary variables has made the solution process more efficient for Model-1.

Mann–Whitney U test is employed for model parameters and performance measures to assess the statistical significance of performance differences. As demonstrated in Table V, Probability (P) values indicate a statistically significant difference between Model-1 and Model-2 in terms of the number of constraints, integer, binary variables, and CPU time under a predefined α level ($\alpha = 0.05$).

Considering the scheduling horizon in a time-indexed model has a great impact on optimum solutions and computational requirements. This means that to obtain an optimal solution, the number of constraints and the CPU time grow exponentially as the time horizon increases, as seen in Fig. 4.

V. CONCLUSIONS

This work commenced by addressing the FJSP that incorporates both site routing and sequencing flexibility (FJSP-SSF), representing a significant class of practical problems. Then two MILP models are developed for the FJSP-SSFs. One is a time-indexed model called Model-1, and another is a precedence-based model, namely Model-2. They are compared by ten FJSP-SSF problems in terms of model size, CPU time, and C_{max}. The findings in the experiments indicate that Model-2 has a significant performance superiority in small-scale problems. As the number of time horizons increases, the number of constraints and CPU time of solving for Model-1 grow rapidly. However, if the appropriate number of time horizons is chosen, Model-1 may outperform Model-2 on medium-scale problems.

The authors of this paper are planning a future study to develop an innovative approach to tighten the formulations of FJSP along with site routing and sequence flexibility.

References

[1] S. Peng, G. Xiong, Y. Ren, Z. Shen, S. Liu, and Y. Han, "A parallel learning approach for the flexible job shop scheduling problem," IEEE Journal of Radio Frequency Identification, vol. 6, pp. 851–856, 2022.

[2] L. Yu, C. Zhu, J. Shi, W. Zhang et al., "An extended flexible job shop scheduling model for flight deck scheduling with priority, parallel operations, and sequence flexibility," Scientific Programming, vol. 2017, 2017.

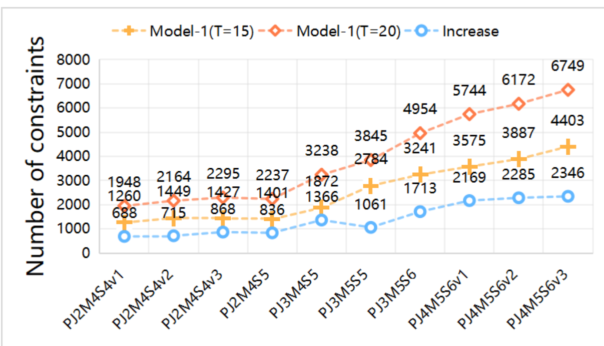
TABLE IV: The computational results of Model-1 and Model-2

Problem No	Model-1						Model-2					
	Constraints	Integer	Binary	NB/I	CPU time(s)	C_{max}	Constraints	Integer	Binary	NB/I[a]	CPU time(s)	C_{max}
PJ2M4S4v1	1260	257	257	0%	18.31	9	602	108	76	30%	5.04	10
PJ2M4S4v2	1449	306	306	0%	39.19	9	616	108	76	30%	2.98	9
PJ2M4S4v3	1427	264	264	0%	19.05	12	770	128	92	28%	5.13	13
PJ2M4S5	1401	266	266	0%	15.84	11	775	116	80	31%	5.80	11
PJ3M4S5	1872	422	422	0%	161.25	12	1421	222	168	24%	49.50	11
PJ3M5S5	2117	503	503	0%	55.54	13	1590	246	180	27%	24.74	13
PJ3M5S6	3241	837	837	0%	132.14	10	2256	270	192	29%	364.58	10
PJ4M5S6v1	3575	926	926	0%	423.91	11	3098	381	285	25%	715.93	11
PJ4M5S6v2	3887	1028	1028	0%	145.97	12	3472	424	320	25%	937.36	12
PJ4M5S6v3	4403	1118	1118	0%	778.68	12	3598	424	320	25%	3600	13

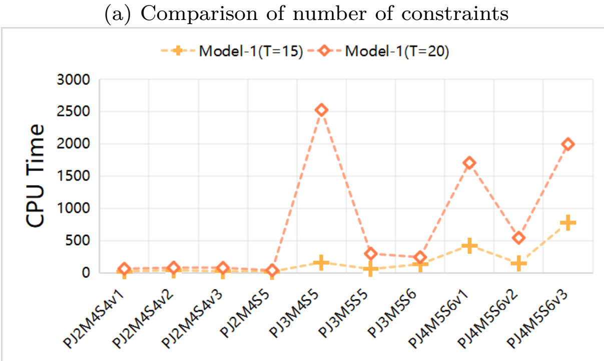
[a] NB/I denotes the ratio of non-binary and integer variables.

TABLE V: Mann–Whitney U test P-values for models

	Constraints	Integer	Binary	CPU time	Cmax
Small	0.0324	0.0024	0.0016	0.0465	0.4673
Medium	0.0196	0.0147	0.0147	0.0196	0.4404



(a) Comparison of number of constraints



(b) Comparison of CPU time

Fig. 4: Computational requirements for different horizons

[3] S. Peng, G. Xiong, J. Yang, Z. Shen, T. S. Tamir, Z. Tao, Y. Han, and F.-Y. Wang, "Multi-agent reinforcement learning for extended flexible job shop scheduling," Machines, vol. 12, no. 1, p. 8, 2023.

[4] Y. Demir and S. K. İşleyen, "Evaluation of mathematical models for flexible job-shop scheduling problems," Applied Mathematical Modelling, vol. 37, no. 3, pp. 977–988, 2013.

[5] P. Fattahi, M. Saidi Mehrabad, and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems," Journal of Intelligent Manufacturing, vol. 18, pp. 331–342, 2007.

[6] P. Fattahi, F. Jolai, and J. Arkat, "Flexible job shop scheduling with overlapping in operations," Applied Mathematical Modelling, vol. 33, no. 7, pp. 3076–3087, 2009.

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

[8] M. Ziaee, "A mixed integer linear programming model for flexible job shop scheduling problem," International Journal of Mathematical and Computational Sciences, vol. 12, no. 6, pp. 95–99, 2018.

[9] C. S. Thomalla, "Job shop scheduling with alternative process plans," International Journal of Production Economics, vol. 74, no. 1-3, pp. 125–134, 2001.

[10] M. Gomes*, A. Barbosa-Povoa, and A. Novais, "Optimal scheduling for flexible job shop operation," International Journal of Production Research, vol. 43, no. 11, pp. 2323–2353, 2005.

[11] L. Meng, C. Zhang, X. Shao, and Y. Ren, "Milp models for energy-aware flexible job shop scheduling problem," Journal of Cleaner Production, vol. 210, pp. 710–723, 2019.

[12] A. Vital-Soto, A. Azab, and M. F. Baki, "Mathematical modeling and a hybridized bacterial foraging optimization algorithm for the flexible job-shop scheduling problem with sequencing flexibility," Journal of Manufacturing Systems, vol. 54, pp. 74–93, 2020.

[13] S. M. Homayouni and D. B. Fontes, "Production and transport scheduling in flexible job shop manufacturing systems," Journal of Global Optimization, vol. 79, no. 2, pp. 463–502, 2021.

[14] H. Zhang and M. Huang, "Multi-attribute graph based mixed integer linear programming model for a job shop scheduling problem considering intelligent vehicle transport," in 2022 International Conference on Machine Learning, Control, and Robotics (MLCR). IEEE, 2022, pp. 62–66.

[15] Y.-J. Yao, Q.-H. Liu, X.-Y. Li, and L. Gao, "A novel milp model for job shop scheduling problem with mobile robots," Robotics and Computer-Integrated Manufacturing, vol. 81, p. 102506, 2023.

[16] A. Liu, P. B. Luh, B. Yan, and M. A. Bragin, "A novel integer linear programming formulation for job-shop scheduling problems," IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 5937–5944, 2021.

[17] B. Yan, M. A. Bragin, and P. B. Luh, "An innovative formulation tightening approach for job-shop scheduling," IEEE Transactions on Automation Science and Engineering, vol. 19, no. 3, pp. 2526–2539, 2021.

[18] A. Liu, P. B. Luh, K. Sun, M. A. Bragin, and B. Yan, "Integrating machine learning and mathematical optimization for job shop scheduling," IEEE Transactions on Automation Science and Engineering, 2023.

[19] J. K. Lenstra, A. R. Kan, and P. Brucker, "Complexity of machine scheduling problems," in Annals of Discrete Mathematics. Elsevier, 1977, vol. 1, pp. 343–362.