



New formulation for scheduling flexible flow shop problems

Majid Esmaelian¹ · Milad Mohammadi¹ · Hadi Shahmoradi¹

Received: 1 August 2021 / Accepted: 14 June 2023 / Published online: 15 July 2023
© The Author(s), under exclusive licence to Springer-Verlag France SAS, part of Springer Nature 2023

Abstract

This research extends the constrained vehicle routing problem concept to solve flexible flow shop scheduling problems. Mixed-integer linear programming and constraint programming formulations are developed for a flow shop problem with no-wait, time lags and release time restrictions to minimize the makespan in both permutation and non-permutation schedules. The comparative analysis of various models reveals that constraint programming models have superior computational performance than mixed-integer linear programming models. However, the mixed-integer linear programming models are also timely-efficient. Moreover, the efficiency of developed models is also represented in comparison with several benchmark datasets. Based on the findings, while the objective function values of the mixed-integer linear programming and constraint programming models in non-permutation schedules exhibit lower values than their respective equivalents in permutation schedules, both models demonstrate longer runtime in non-permutation schedules. Results represent that the proposed constraint programming and mixed-integer linear programming models are among the top three models of the benchmark datasets in terms of the number of decision variables and computational performance. One of the limitations of the research is that there is no comprehensive dataset in the literature considering all the restrictions in permutation and non-permutation schedules.

Keywords Flexible flow shop · Mixed integer linear programming · Constraint programming · Constrained vehicle routing problem

1 Introduction

Production and operations scheduling is one of the most significant factors that may lead to the failure or success of a manufacturing company [1, 2]. The job scheduling problem is regarded as arranging operation tasks (jobs) in a production system and focuses on specifying the orders of jobs on a pre-defined set of machines. These job orders are usually determined so that the operational or financial performance of the given production system is optimized. The job scheduling problem is known as the flow shop problem when all jobs have the same sequence on machines, and each job requires only one operation on each machine [3]. The flexible flow shop (FFS) problem is a generalization of the flow shop where

every job can be processed by one of the machines in each station [4]. In the FFS problem, the production system has a fixed sequence of stations, and each station includes a set of machines. So far, FFS problems are widely applied in different industries such as chemical, logistics and semiconductors. [5].

In the related literature, flow shop problems are usually categorized into two main groups: Cyclic and Non-cyclic [3, 6, 7]. Each category considers different production settings (assumptions) such as no-wait [8], time lags [9], blocking [10, 11], and limited or unlimited intermediate storage levels [12]. In no-wait flow shop problems, each job must be processed without interruption between machines [13, 14]. In the flow shop problems with time lags, practitioners consider the minimum and maximum wait time between every two consecutive operations which are required for completing a job on machines [15–17]. Both cyclic and non-cycle flow shop problems can be formulated to provide permutation and non-permutation schedules for jobs. In permutation flow shop scheduling, the sequence of jobs must be the same on all machines [18, 19]. However, this sequence can be different when the flow shop problem considers non-permutation orders for jobs. [20, 21].

✉ Majid Esmaelian
m.esmaelian@ase.ui.ac.ir
Milad Mohammadi
m.mohammadi@ase.ui.ac.ir
Hadi Shahmoradi
ha.shahmoradi@ase.ui.ac.ir

¹ Department of Management, University of Isfahan, Hezarjerib St., Azadi Square, Isfahan 7344181746, Iran

From the computational point of view, flow shop problems are usually NP-hard combinatorial optimization problems resulting in high dimension solution regions with many locally optimal solutions [22]. Therefore, scholars employ different heuristics and non-heuristics modeling methods to estimate the optimal or near-optimal schedules for the production [23]. Mixed Integer Programming (MILP) and Constraint Programming (CP) are recognized as mature optimization techniques to solve both NP-hard and non-NP-hard problems. Both techniques may enable users to model scheduling problems and reach specific values for their decision variables, known as "exact" optimal solutions [24–26]. CP is a programming paradigm that models the relation between decision variables in the form of constraints [26]. This technique, originated from artificial intelligence, aims to efficiently estimate optimal solutions using constraints in an optimization problem [27]. In general, CP models have several advantages in solving optimization problems: 1) CP models can find infeasibilities immediately, 2) CP models can obtain the initial feasible solutions quickly, 3) CP models can obtain the optimal solution in low computational times, and 4) CP languages assist in the model development because they are declarative [28]. Due to these advantages, a remarkable trend of attention has been attracted to the CP models, particularly for scheduling problems.

FFS problems have received increasing research attempts during the last decade due to their real-world application in manufacturing industries. However, some notable aspects of the FFS problem have received less attention. Some of the most related studies in this area are reviewed in the following. Mucha and Sviridenko [8] worked on the classical problem of no-wait flow shop scheduling with makespan objective and considered this problem as a particular case of the asymmetric traveling salesman problem. Dorfeshan et al. [9] considered transportation lags and a weighted distance-based approximation method to arrange jobs in flow shop scheduling problems. Rossit et al. [29] presented a combinatorial analysis of the permutation and non-permutation flow shop scheduling problems with unknown processing times. A systematic literature review on permutation and non-permutation flow shop problems is provided by Singh et al. [30]. They worked on the multi-objective flexible flow shop scheduling with no-wait constraints. Samarghandi [31] developed constraint programming and MILP models for a flow shop environment under minimum and maximum time-lag constraints in which the makespan is minimized. Furthermore, Qin et al. [32] studied the integrated CP and mixed-integer programming to solve the hybrid flow shop scheduling problem. Xiao et al. [33] formulated the non-permutation flow shop scheduling problem with order acceptance as a MILP model. Moreover, Xin et al. [34] proposed a MILP model for the flow shop scheduling problem with sequence-dependent setup time to minimize the

makespan and total energy consumption. Table 1 demonstrates methods, objective functions, and features of flow shop problems considered in the literature to schedule jobs.

The main contribution of the present study is to extend the CVRP formulations for FFS with no-wait, time lags, and release time restrictions and the makespan (C_{\max}) function in both permutation and non-permutation schedules. In addition to presenting MILP models for different FFS problems, the constraint programming (CP) models corresponding to the underlying FFS problems are also proposed, and the results of developed models are compared. However, the literature review reveals that scholars usually apply MILP methods for specific flow shop scheduling problems. Their MILP formulations were developed to consider only permutation or non-permutation schedules for given flow shop problems. In other words, their MILP methods cannot be applied to solve both permutation and non-permutation scheduling problems. Developing MILP formulation according to Capacitated Vehicle Routing Problem (CVRP) concept would be one of the modeling approaches to remove the above limitations in creating a generalized method to solve flow shop job scheduling problems. To the best knowledge of the authors, there is no such modeling approach for flexible flow shop problems.

2 Proposed mixed-integer linear programming model

The FFS problem considered in this study is described as a set of n jobs (tasks) that should be processed on a set of stations ($l = 1, \dots, c$) containing $M_l \geq 1$ identical parallel machines. A function $\pi_i = [p_i^1, p_i^2, \dots, p_i^c]$ demonstrates the required processing times of the i^{th} job in different stations. Each job can be processed by any of the M_l parallel machines in l^{th} station (Fig. 1). The FFS problem is solved to find a sequence of n jobs that should be processed so that some objectives such as the total required time for completing all jobs, the maximum lateness, or the tardy job count are optimized. The MILP models of the present study are generalized based on the CVRP formulation for the FFS problem with no-wait, time lags, and release time restrictions on time-capacitated machines with permutation and non-permutation schedules.

CVRP is defined on a graph $G = (N, D)$ where $N = \{0, 1, 2, \dots, n\}$ is the vertex (node) set and D is the edge (arc) set where $D = \{(i, j) : i, j \in N, i \neq j\}$. Vertex i ($i = 1, \dots, n$) represents customers, while vertex 0 is defined as the depot (dummy) node. This node is considered as the depot of m identical vehicles with a specific capacity level. Each customer i ($i = 1, \dots, n$) is associated with a positive integer demand, and each set of (i, j) edge is associated with a travel cost or travel time. The CVRP aims to determine a set of (at

Table 1 Common features of articles in flow shop problems

Authors	Cyclic	Non-cyclic	No-wait	Blocking	Unlimited-buffer	Limited-buffer	Time lags	Permutation(P) /Non-permutation(NP)	Objective function	Method
[35]			*		*			NP	Total order earliness	MILP
[36]			*				*	P	Makespan	Branch and bound procedure
[37]			*		*			P	Makespan	MILP
[38]			*				*	P	A non-classical criterion based on the weighted sum of machine completion times	An exact algorithm based on a branch-and-bound procedure
[39]		*			*		*	P	Makespan	MIP
[40]				*		*		NP	Makespan	Quadratic MIP and approximation algorithm
[41]			*	*			*	NP	Total weighted tardiness	MILP
[42]	*			*				NP	Cycle time	MIP
[43]				*		*	*	NP	Makespan	MILP and GA algorithm
[44]	*		*		*			NP	Annual investment cost of batch plant,	MILP
[45]			*	*				P	Summation of weighted earliness and tardiness, considering (SDSC)	MILP, TS and PSO
[46]	*			*				NP	Cycle time	Branch and Bound algorithm
[47]	*					*		NP	Cycle time	MILP
[48]		*	*		*			P	Total Flow Time (TFT)	MILP and HGA-VNS
[14]	*		*					NP	Cycle time	MILP
[49]				*				P	Makespan	MILP, ACO and GA
[7]		*			*			P	Makespan	Branch and Bound algorithm
[50]			*					P	Makespan	One MILP, two quadratic and two CP models
Present study		*	*		*		*	P and NP	Makespan	MILP & CP

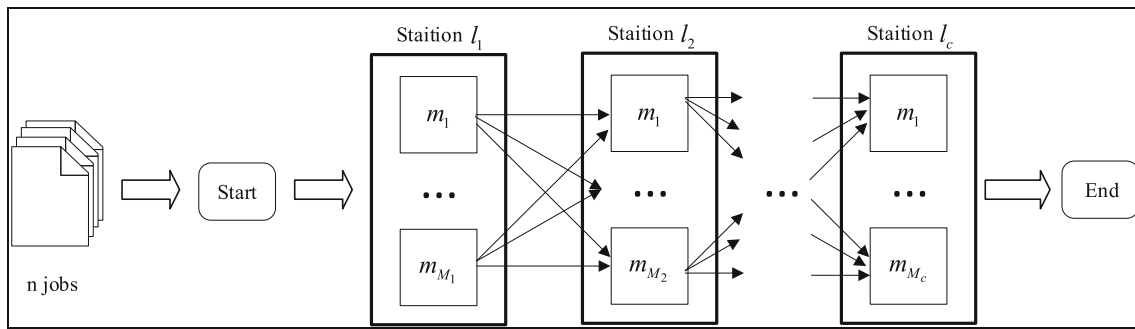


Fig. 1 A flexible flow shop layout

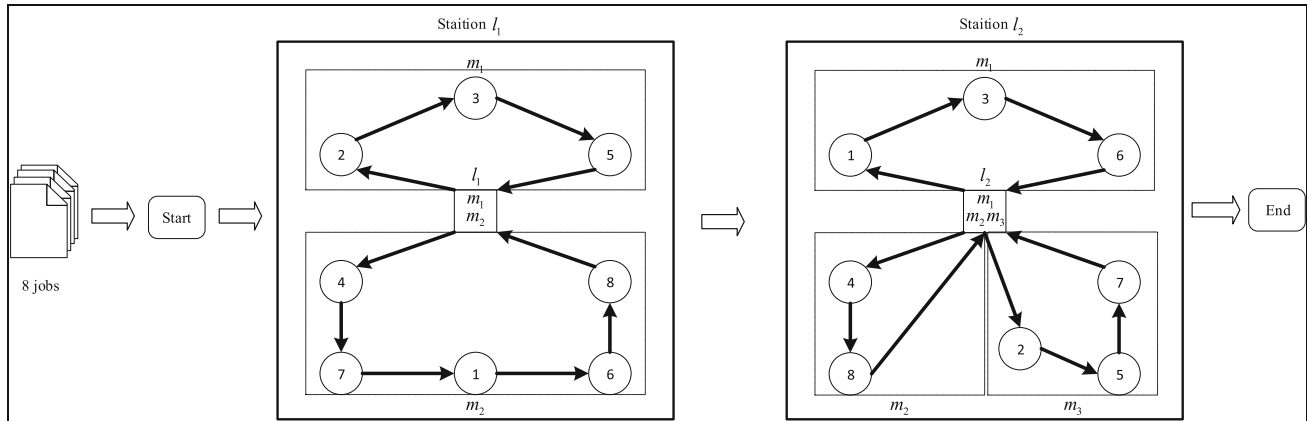


Fig. 2 A simple correspondence of FFS with CVRP

most) m vehicle routes such that: a) each route must start and end at the depot vertex (node 0), b) each customer (vertex) must belong to exactly one route, c) the total load on each route does not exceed the capacity level of the vehicle, and, finally, d) the total cost (or time) of all m routes is minimized. FFS problem with n jobs, c stations, and M_l machines in l^{th} station is equivalent to c successive CVRP models with n customers and M_l time-capacitated machines in l^{th} station. Figure 2 shows a simple adoption of non-permutation FFS with two successive CVRPs, scheduling eight jobs processed on two stations with two and three identical parallel machines, respectively.

The following assumptions are considered in formulating the FFS problem: a) All machines in each station are parallel and identical with the same capabilities in job processing; b) If a job is assigned to a machine, the entire job is completely accomplished on this machine. There exist no possibility of breaking a job and accomplishing a part of it in another time on the same machine. c) Each machine is capable of completing the processing of a job in a given time; d) Each job can be processed after completing the other job, with no precedence constraints, e) All machines are available through the scheduling process and can continuously perform operations at time zero, and f) The processing time of each job on each station is pre-determined.

To specify scheduling problems economically and precisely, Graham et al. [51] introduced a shorthand notation. The classification system comprises three fields separated by bars: $\alpha|\beta|\gamma$. In the field α , the type and size of the shop are entered. For example, FF_c denotes the flexible flow shop with c stages (or stations). In the field β , the special features of the shop are listed. For example, $prmu$ represents permutation schedules (where the sequence of job processing on all stages is the same), r represents the release time restriction (the earliest time that a job is available to be processed), l represents the lag time restriction (the possibility that an additional time delay should elapse between completing a job at one station and starting it at the next), nwt represents the no-wait restriction (where a job, once started, must flow through every station to completion without any delay). In the field γ , the objective function or criterion that should be maximized or minimized is entered. For example, C_{max} denotes the maximal or latest completion time of any job (this criterion commonly is called the makespan) [52].

In this study, the CVRP formulations for FFS with no-wait ($FF_c|nwt|C_{max}$, $FF_c|nwt, prmu|C_{max}$), time lags ($FF_c|l|C_{max}$, $FF_c|l, prmu|C_{max}$) and release time restrictions ($FF_c|r|C_{max}$, $FF_c|r, prmu|C_{max}$) and C_{max} function in both permutation and non-permutation schedules are

extended. Minimizing the makespan (C_{\max}) is the objective function of the developed optimization models. The constraint programming (CP) models corresponding to the underlying FFS problems are also proposed.

2.1 Parameter

n : Number of jobs ($i = 1, \dots, n$)
 c : Number of stations ($l = 1, \dots, c$)
 p_i^l : Processing time of i^{th} job in l^{th} station

M_l : The number of machines in l^{th} station.

R_i : Release time of i^{th} job.

H : A large constant value ($H \rightarrow \infty$).

In the CVRP model, a dummy job ($i = 0$) with zero processing time is added to the list of jobs, which determines the beginning and the end of the turn in the CVRP problem. (expressed as $i = 0, 1, \dots, n$).

2.2 Variables

x_{ij}^l : If the j^{th} job is performing immediately after the i^{th} job in l^{th} station (1), otherwise (0).

v_i^l : Cumulative time of the jobs processed on each machine until completion of the processing of the i^{th} job in the l^{th} station. Variable v_i^l is defined to eliminate sub-tours in each station.

u_i^l : Cumulative time of the jobs processed on each machine until completion of the i^{th} job from first to the l^{th} station. The completion time of the i^{th} job in the l^{th} station is determined by u_i^l . For the last station, u_i^c is equivalent to the flow time completion of i^{th} job. Therefore, the makespan or maximum flow time of jobs in the last station is calculated as $C_{\max} = \max_{i=1}^n (u_i^c)$.

s_i : Start time of jobs' processing in the first station. In the presented model, it is unnecessary to start processing jobs on the first station at time zero. The variable s_i is defined for all jobs at the first station, and if i^{th} job is the first job on any of the machines, then the start time of processing of this job ($u_i^1 - p_i^1$) is calculated from time s_i as $u_i^1 - p_i^1 = s_i$.

Some pre-defined time lags exist between the same jobs in successive stations in the flow shop problem with time lag restriction. Processing of i^{th} job is completed at u_i^1 in station one, and it should be resumed at ($u_i^2 - p_i^2$) in station two after elapsing time lag ($lag_i^1 = u_i^2 - p_i^2 - u_i^1$).

2.3 MILP model

The MILP model for $FFC|C_{\max}$ (FFS non-permutation model) is formulated as follows:

2.3.1 Non-Permutation MILP model (MILP-NP)

$$\text{Minimize } C_{\max} \quad (1)$$

$$\sum_{i=1}^n x_{0i}^l = \sum_{i=1}^n x_{i0}^l = M_l; \quad l = 1, \dots, c \quad (2)$$

$$\sum_{\substack{j=0 \\ i \neq j}}^n x_{ij}^l = \sum_{\substack{j=0 \\ i \neq j}}^n x_{ji}^l = 1; \quad l = 1, \dots, c; \quad i = 1, \dots, n \quad (3)$$

$$v_i^l - v_j^l + Hx_{ij}^l + (H - p_i^l - p_j^l)x_{ji}^l \leq H - p_j^l; \quad l = 1, \dots, c; \quad 1 \leq i \neq j \leq n \quad (4)$$

$$v_i^l \geq p_i^l; \quad l = 1, \dots, c; \quad i = 1, \dots, n \quad (5)$$

$$v_i^l + (H - p_i^l)x_{0i}^l \leq H; \quad i = 1, \dots, n; \quad l = 1, \dots, c \quad (6)$$

$$u_i^l - p_i^l \geq u_i^{l-1}; \quad l = 2, \dots, c; \quad i = 1, \dots, n \quad (7)$$

$$u_i^1 + (H - p_i^1)x_{0i}^1 \leq H; \quad i = 1, \dots, n \quad (8)$$

$$u_i^1 \geq p_i^1; \quad i = 1, \dots, n \quad (9)$$

$$u_j^l - p_j^l \geq u_i^l - H(1 - x_{ij}^l); \quad l = 1, \dots, c; \quad 1 \leq i \neq j \leq n \quad (10)$$

$$C_{\max} \geq u_i^c; \quad i = 1, \dots, n \quad (11)$$

$$u_i^l, v_i^l \geq 0; \quad i = 1, \dots, n; \quad l = 1, \dots, c \quad (12)$$

$$x_{ij}^l \in \{0, 1\}; \quad 0 \leq i \neq j \leq n; \quad l = 1, \dots, c \quad (13)$$

Equation (1) is the objective function that minimizes the makespan. Equation (2) assures that each turn (machine) in the first station starts and ends with the dummy node 0. Equation (3) is the usual balancing constraint for CVRP. Equation (4) assures the sub-tour elimination for each machine in each station. This inequality for all jobs $1 \leq i \neq j \leq n$ ensures that $v_i^l = v_j^l + p_j^l$ only if $x_{ij}^l = 1$ and $v_j^l = v_i^l + p_i^l$ only if $x_{ji}^l = 1$. In the original CVRP model, the variable v_j^l is computed cumulatively and continuously; in other words, if the j^{th} job is processed immediately after the i^{th} job, then $v_j^l = p_j^l + v_i^l$. This computation is not valid for the cumulative time because in the optimal scheduling solution, for two successive jobs, the start time of the following job is not immediately after the finish time of the before job, and it is possible to have idle time between two jobs. The variable v_j^l does not have this condition, while by defining the variable

u_i^l and adding Eq. (5)-(10), the start and finish time of successive jobs on a machine is computed correctly. Equation (11) ensures that the makespan must be greater than (or equal to) the cumulative time of the jobs processed on the last station. Equation (12) and Eq. (13) assure the positive and binary conditions of variables, respectively.

By converting Eq. (7) into $u_i^l - p_i^l = u_i^{l-1}$, the MILP model for $FFC|nwt|C_{max}$ (no-wait FFS model) is formed. The FFS will be formed with minimum time lags by adding the parameter Lag_i^l to Eq. (7) as $u_i^l - p_i^l \geq u_i^{l-1} + Lag_i^l$. By converting this inequality to $u_i^l - p_i^l = u_i^{l-1} + Lag_i^l$, the model for FFS with time lags restriction and by adding $u_i^l - p_i^l \leq u_i^{l-1} + Lag_i^l$ to the model, the FFS with maximum time lags is formulated.

Through changing Eq. (8) and Eq. (9) to $u_i^1 + (H - p_i^1)x_{0i}^1 - s_i \leq H$ and $u_i^1 \geq p_i^1 + s_i$, and adding Eq. (14), Eq. (15) and Eq. (16), the MILP model for FFS with the release time restriction ($FFC|r|C_{max}$) is formed.

$$s_i \leq Hx_{0i}^1; \quad i = 1, \dots, n \quad (14)$$

$$u_i^1 - p_i^1 \geq R_i; \quad i = 1, \dots, n \quad (15)$$

$$s_i \geq 0; \quad i = 1, \dots, n \quad (16)$$

The MILP model can be extended for $FFC|prmu|C_{max}$ (FFS with permutation). In the permutation schedule, the job permutation in all stations is the same. Therefore, the variable x_{ij} is defined as if the j^{th} job performs immediately after i^{th} job (1), otherwise (0).

In the permutation schedule, sub-tours elimination is only needed in the first station. Therefore, variable v_i is defined as a cumulative time of the jobs processed on each machine until completion of the processing of the i^{th} job in the first station. In the permutation schedule, the number of machines in all stations is equal. The MILP model for $FFC|prmu|C_{max}$ is formulated as follows:

2.3.2 Permutation MILP model (MILP-P)

$$\text{Minimize } C_{max} \quad (17)$$

$$\sum_{i=1}^n x_{0i} = \sum_{i=1}^n x_{i0} = M \quad (18)$$

$$\sum_{\substack{j=0 \\ i \neq j}}^n x_{ij} = \sum_{\substack{j=0 \\ i \neq j}}^n x_{ji} = 1; \quad i = 1, \dots, n \quad (19)$$

$$v_i - v_j + Hx_{ij} + (H - p_i^1 - p_j^1)x_{ji} \leq H - p_j^1; \quad 1 \leq i \neq j \leq n \quad (20)$$

$$v_i \geq p_i^1; \quad i = 1, \dots, n \quad (21)$$

$$v_i + (H - p_i^1)x_{0i} \leq H; \quad i = 1, \dots, n \quad (22)$$

$$u_i^l - p_i^l \geq u_i^{l-1}; \quad l = 2, \dots, c; \quad i = 1, \dots, n \quad (23)$$

$$u_i^1 + (H - p_i^1)x_{0i} \leq H; \quad i = 1, \dots, n \quad (24)$$

$$u_i^1 \geq p_i^1; \quad i = 1, \dots, n \quad (25)$$

$$u_j^l - p_j^l \geq u_i^l - H(1 - x_{ij}); \quad l = 1, \dots, c; \quad 1 \leq i \neq j \leq n \quad (26)$$

$$C_{max} \geq u_i^c; \quad i = 1, \dots, n \quad (27)$$

$$u_i^l, v_i \geq 0; \quad i = 1, \dots, n; \quad l = 1, \dots, c \quad (28)$$

$$x_{ij} \in \{0, 1\}; \quad 0 \leq i \neq j \leq n \quad (29)$$

The description of Eq. (17)-(29) is the same as Eq. (1)-(13), respectively. The only difference between permutation and non-permutation models is the definition of x_{ij} and v_i variables (explained at the end of Sect. 2.3.1). No-wait, time lags, and release time restrictions can be considered in the model for $FFC|prmu|C_{max}$ as described for the model $FFC|C_{max}$.

The proposed FFS with different restrictions can be modeled through CP properties. The CP models differ from the corresponding MILP representation and are described in Sect. 3.

3 Constraint programming models

Constraint programming is a tool for solving combinatorial search problems and has been addressed in various techniques such as artificial intelligence, computer science, and operations research. Earlier, CP models were used to assign values from the variables' domain which satisfies all the constraints of a constraint satisfaction problem. Later, CP search procedures were extended to efficiently estimate optimal solutions using constraints in a constraint optimization problem [27]. In this part, some notable features of the CP in both problem formulation and solving algorithms are mentioned. First of all, CP can deal with different types of decision variables such as interval and sequence-based variables and also integer variables [53]. Second, CP benefits from general constraints to develop models compactly [26]. Third, CP models have no limitation using different types of problem constraints and objective functions (e.g., linear, nonlinear,

convex), which are challenging in developing MILP models. Moreover, from solving algorithms points of view, CP usually uses the propagation technique to remove unfeasible regions of the search space [54, 55]. CP benefits from this technique to tackle large optimization problems that may not be solvable by MILP models in a reasonable time frame [56].

In the present study, the Optimization Programming Language (OPL) and IBM ILOG CP optimizer platform are used to model and solve the problem. OPL is a modeling language for combinatorial optimization, designed to simplify optimization problems substantially. This language increases the applicability of modeling languages by incorporating techniques from constraint programming. The remainder of this part discusses how to formulate the CP models.

3.1 Variables

Two types of variables are defined:

1. Interval variables: these variables represent an interval of time during which something happens (a task or an activity is carried out). An interval is characterized by beginning, ending, and size values, which OPL offers some functions to retrieve these features. Interval variables are optional; that is, one can decide not to consider them in the solution schedule. According to this definition, an optional interval variable x_{im}^l is defined, representing the i^{th} job on m^{th} machines in l^{th} station ($i = 1, \dots, n, m \in M_l, l = 1, \dots, c$).
2. Interval sequence variable: these variables represent a total order over a set of interval variables. In this newly proposed model, there exist some stations which can be mapped as sequence variables. Also, each of these variables can be divided into some sub-sequence variables that represent the machines in each station. Every station includes all the interval variables distributed on its machines. According to this definition, an interval sequence variable q_m^l is defined, which represents the m^{th} machine in l^{th} station ($m \in M_l$).

3.2 CP model

This section presents the corresponding CP models for $FF_c|C_{max}$ (FFS system with permutation) and $FF_c|prmu|C_{max}$ (non-permutation FFS). The required parameters and variables are the same as defined in sub-Sects 2.1 and 3.1, respectively.

3.2.1 Non-Permutation CP model (CP-NP)

The CP model for $FF_c|C_{max}$ is formulated as below:

$$\begin{aligned} \text{Minimize } C_{\max} &= \max(\text{end } Of(x_{im}^c)); \quad i \\ &= 1, \dots, n, \quad m \in M_c \end{aligned} \quad (30)$$

Subject to:

$$\sum_{m \in M_l} \text{presence } Of(x_{im}^l) = 1; \quad i = 1, \dots, n; \quad l = 1, \dots, c \quad (31)$$

$$(q_m^l); \quad \text{for all } m \in M_l, \quad l = 1, \dots, c \quad (32)$$

$$\begin{aligned} \text{start } Of(x_{im_2}^{l_2}) &\geq \text{end } Of(x_{im_1}^{l_1}); \quad i = 1, \dots, n, \quad l_1 = l_2 \\ &= 1, \dots, c, \quad m_1 \in M_{l_1}, \quad m_2 \in M_{l_2} | l_2 > l_1 \end{aligned} \quad (33)$$

$$\begin{aligned} L_m &\leq \max(\text{end } Of(x_{im}^l)) \leq U_m; \quad i \\ &= 1, \dots, n, \quad l = 1, \dots, c, \quad m \in M_l \end{aligned} \quad (34)$$

The objective function is defined as the maximum finish time of jobs in the last station and expressed as Eq. (30). Due to the optionality feature of interval variables, a function (called *presenceOf*) could be used to indicate whether an interval variable is absent or present. By considering this, Eq. (31) is defined to assure that all jobs are carried out in each station, and only one job appears on only one machine in each station. OPL has some pre-defined functions which indicate the flexibility of the CP model. One of them is *noOverlap*, which guarantees that there is no overlap between any two interval variables which exist in a sequence. This constraint is applied to interval sequence variables, expressed as Eq. (32). This constraint states that the machines in all stations define a chain of non-overlapping jobs. Equation (33) assures that the same variable placed in the successive stations starts after the previous stations. Moreover, Eq. (34) is applied to ensure the upper and lower bounds of the time capacity of each machine.

By converting Eq. (33) to $\text{start } Of(x_{im_2}^l) == \text{end } Of(x_{im_1}^{l-1})$, the CP model for $FF_c|nwt|C_{max}$ is formed. Adding the parameter Lag_i^l to the Eq.(33) as $\text{start } Of(x_{im_2}^l) \geq \text{end } Of(x_{im_1}^{l-1}) + Lag_i^l$, the FFS will be formed with minimum time lags. Adding the parameter Lag_i^l to the Eq. (33) as $\text{start } Of(x_{im_2}^l) == \text{end } Of(x_{im_1}^{l-1}) + Lag_i^l$, the FFS will be formed with a time lag restriction. Finally, by adding $\text{start } Of(x_{im_2}^l) \leq \text{end } Of(x_{im_1}^{l-1}) + Lag_i^l$ to the model, the FFS with maximum time lags is formulated.

By adding $\text{start } Of(x_{im}^1) \geq R_i$, the CP model for FFS with release times restriction, $FF_c|r|C_{max}$, is formed. This

constraint states that the processing of all jobs in the first station occurs after the released time (R_i).

3.2.2 Permutation CP model (CP-P)

CP model for $FF_c|C_{max}$ can be extended to $FF_c|prmu|C_{max}$ by adding a constraint to assure that every machine in all stations has the same job permutation. Equation (35) provides such that constraint. The OPL function *type Of Next* (q_m^l, x_{im}^l) in this constraint refers to the following interval variable from x_{im}^l in machine q_m^l .

$$\begin{aligned} \text{type Of Next}(q_{m_1}^{l_1}, x_{im_1}^{l_1}) &= \text{type Of Next}(q_{m_2}^{l_2}, \\ &x_{im_2}^{l_2}); \quad i = 1, \dots, n; \\ &m_1 \in M_{l_1}; \quad m_2 \in M_{l_2} \end{aligned} \quad (35)$$

No-wait, time lags, and release time restrictions can be considered in $FF_c|prmu|C_{max}$ as described for model $FF_c|C_{max}$.

4 Results of the models and efficiency comparison

For computational analysis purposes, several test problems are selected from the literature available for $FF_c|nwt, prmu|C_{max}$. These test problems include *Car* problems (Car01-08) [57] and *Rec* problems (Rec01, 03, 05, 41) [58], which are available in the OR-Library.¹ *Car* problems generally contain a small number of jobs and stations (Sect. 4.1), while *Rec* problems correspond to real dimensions of industrial problems, which contain many jobs and stations (Sect. 4.2).

4.1 Problems with optimal solutions for ($FF_c|nwt, prmu|C_{max}$)

Conducting numerical experiments is a practical approach to compare the performance of the developed models. To verify the performance of the proposed models on the test problems, the results of the model obtained in Manne [59] and Samarghandi [50] are considered for comparison purposes. The numerical results of the following models are considered: 1) No-wait version of the Manne model; Manne model is one of the best and most commonly used models, 2) Original formulation of Model I of Samarghandi at relaxed due date constraints, 3) Model I of Samarghandi with indicator constraints and without due date constraints, 4) Model II of Samarghandi, which includes two dummy jobs that should

be located in the first and the last positions in the sequence, 5) Model IV of Samarghandi, unlike previous models, is formulated based on properties of CP, 6) Model V of Samarghandi, is formulated based on CP. Due date constraints are relaxed in Model II, IV, and V. The newly proposed models of MILP-P and CP-P constitute the last two columns of Table 2.

The large number (H) is applied in the formulation of MILP models for either-or constraints. Although this is a proper manner to prototype either-or constraints, the numerical value of H may result in complexity in running the model in IBM CPLEX and any other software package. If the value of H is not selected accurately, the CPLEX may eliminate H in the pre-solve phase. Consequently, it is recommended² that either-or constraints should be formulated through indicator constraints instead of implementing the large number H . Whereas, applying indicator constraints would decrease the effectiveness of the branching algorithm, which in turn could increase the solution time. Based on solving different models through indicator constraints and the large number H , it was found that considering $H = \sum_{i=1}^n p_i^l$; $l = 1, \dots, c$ in constraints of each station is more efficient. Therefore, the large number H is applied in solving MILP models.

The IBM ILOG CPLEX V12.3 is applied to solve the newly developed MILP and CP models (maximum time limit 600Sec). All the numerical experiments are run on a PC with CPU- Intel Cori 7. 1.6 GHz and RAM-6 GB. In Table 2, OFV represents the objective function value and all of the CPU times are represented in seconds. The time when the optimal solution is found is reported. For instance, the optimal solution of *Car03* is 8866; the MILP-P finds this solution after 173 s and the CP-P after 30 s. Instances without time show the objective function value at the maximum allowed CPU time (600Sec). The figures in boldface demonstrate the optimal solution in the time limit.

Based on Table 2, in all eight studied problems, the proposed MILP and CP models of the present study achieve the optimal solution. From the runtime point of view, MILP Model II of Samarghandi & Behrooz (2017), the proposed CP, and MILP models are ranked first, second, and third, respectively. Since all the models of Samarghandi & Behrooz (2017) are presented for $FF_c|nwt, prmu|C_{max}$ problems, the performance comparison is not possible for $FF_c|nwt|C_{max}$ problems. However, in Table 3, the values of the objective function and runtime are compared in various permutation and non-permutation schedules. The results of the presented models in the non-permutation schedule adopting MILP-NP and CP-NP are tabulated in Table 3. The results of non-permutation models are presented considering three machines in all stations.

Although the objective function values in the MILP and CP models in non-permutation schedules are less than the

¹ <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>

² <http://www-01.ibm.com/support/docview.wss?uid=swg21400084>.

Table 2 Computational results of the $FF_c|nwt, prmu|C_{max}$ problems with the optimal solution

Problem	Size $n \times c$	No-wait OFV*	The result from [50]					MILP-P <i>prmutation</i> with no-wait restrictions	CP-P <i>prmutation</i> with no-wait restrictions
			MILP Manne Model with no-wait constraints	MILP Model I (original formulation)	MILP Model I (indicator variable)	MILP Model II	CP Model IV		
Car07	7*7	7705	7705, 2	7705, 1	7705, 39	7705, 1	7705, 1	7705, 2	7705, 0,9
Car08	8*8	9372	9372, 2	9372, 2	9372	9372, 1	9372, 5	9372, 5	9372, 3,5
Car06	8*9	9690	9690, 2	9690, 2	9690	9690, 1	9690, 2	9690, 5	9690, 4
Car05	10*6	9159	9159, 2	9159, 200	9159	9159, 1	9159, 203	9159, 6	9159, 4
Car01	11*5	8142	8142, 45	8142	8142	8142, 2	8142	8142, 42	8142, 10
Car03	12*5	8866	8866, 180	8866	8866	8866, 6	8866	8866, 173	8866, 30
Car02	13*4	8242	8242, 150	8242	8299	8242, 1	8247	8242, 254	8242, 20
Car04	14*4	9195	9195	9195	9467	9195, 5	9195	9195, 261	9195, 100

Performance comparison of models based on the average value of the objective function and the solution time

Problem	No-wait OFV*	MILP Manne Model with no-wait constraints	MILP Model I (original formulation)	MILP Model I (indicator variable)	MILP Model II	CP Model IV	CP Model V	MILP-P <i>prmutation</i> with no-wait restrictions	CP-P <i>prmutation</i> with no-wait restrictions
Average of objective function value	8796.38	8796.38	8796.38	8837.50	8796.38	8797	15,162.25	8796.38	8796.38
Rank	1	1	1	3	1	2	4	1	1
Average of solution time (s)	-	122.8	325.63	529.88	2.25	326.38	600	93.50	21.55
Rank	-	4	5	7	1	6	8	3	2

*a: OFV: Objective function value

*b: All the OFVs in this column belong to the optimum solution of $FF_c|nwt, prmu|C_{max}$

Table 3 Computational results of the $FF_c|nwt|C_{max}$ for problems with optimal solutions

Problem	Size $n \times c$	Optimum OFV of $FF_c nwt, prmu C_{max}$ (M = 1)	MILP-P <i>prmutation</i> with no-wait restrictions (M = 1)	CP-P <i>prmutation</i> with no-wait restrictions (M = 1)	MILP-NP Non-permutation with no-wait restrictions (M = 3)	CP-NP Non-permutation with no-wait restrictions (M = 3)
Car07	7*7	7705	7705, 2	7705, 0.9	4216, 5	4216, 5
Car08	8*8	9372	9372, 5	9372, 3.5	5093,105	5093,121
Car06	8*9	9690	9690, 5	9690, 4	5486,9	5486,25
Car05	10*6	9159	9159, 6	9159, 4	4526	4406
Car01	11*5	8142	8142, 42	8142, 10	3991	3604
Car03	12*5	8866	8866, 173	8866, 30	4338	3666
Car02	13*4	8242	8242, 254	8242,20	4522	3463
Car04	14*4	9195	9195, 261	9195,100	5117	3687

objective function values of the equivalent model in the permutation schedules, the runtime of both MILP and CP models in the non-permutation schedule is longer than the runtime in permutation schedules. Note that in all the studied Car problems, permutation scheduling models have been solved at the time limit (600Sec), but in five non-permutation schedule problems, the models have not reached the optimal solution within the specified time limit.

4.2 Problems without optimal solutions

$FF_c|nwt, prmu|C_{max}$

A set of 21 *Rec* problems are solved through the proposed models. The results are compared with the results obtained through other studies available in the literature that developed for $FF_c|nwt, prmu|C_{max}$. The computational results of the proposed MILP-P and CP-P for the problems without optimal solutions are tabulated in Table 4. Since the optimal solution cannot be found in less time through MILP-P and CP-P, both models are solved in 1800s. The feasible solution for Rec 37, 39, and 41 in the time limit cannot be found through MILP-P.

In this Table, the first two columns represent the name and size of test problems. The third column presents the makespan of the model of Rajendran [60] for comparison. The next two columns represent the objective function value of the MILP-P and the deviation percentage between the reported objective function and the makespan values of Rajendran [60]. The deviation percentage between the objective function value of these proposed models and the model of Rajendran [60] is computed through Eq. (36):

Deviation Percentage

$$= \left(\frac{OFV^{Method} - OFV^{Rajendran}}{OFV^{Rajendran}} \right) \times 100 \quad (36)$$

In this context, the more negative deviation percentage, the more efficient the method. This fact assures the competitiveness of these proposed models. The next two columns represent the best solution obtained in five runs for the test problem in $FF_c|nwt, prmu|C_{max}$ through the PSO algorithm of Samarghandi [61] and the TS + PSO algorithm (a hybrid of the Tabu search and PSO) of Samarghandi & ElMekkawy [62].

The MILP is compared with the three best models of Manne [59], Wilson [63], and Wagner [64] as mentioned in the study of Pan [65], in addition to three methods introduced by Stafford [66], Šeda [67] and Samarghandi & Behroozi [50] in Table 5.

In Table 6 and Fig. 3, the number of decision variables of different models is compared in terms of different numbers of jobs and stations. The lowest average number of decision variables in all problems belongs to Model I of Samarghandi, and the proposed model of this research is in the second rank with only 1% more variables.

Figure 3 represents that the number of decision variables of the proposed MILP-P model, Model I of Samarghandi and models of Stafford and Seda is much less than models of Wagner, Manne and Wilson.

Moreover, in Table 7, the number of constraints of different models is compared in different numbers of jobs and stations. The lowest average number of constraints in all problems belongs to the Stafford model.

Table 4 Computational results of the $FF_c|nwt, prmu|C_{max}$ for problems without optimal solutions

Problem	Size $n \times c$	Rajendran [60]	MILP-P With No-wait		CP-P With No-wait		(PSO algorithm) Samarghandi [61]		(TS + PSO algorithm) Samarghandi and ElMekkawy [61]	
		OFV	OFV	%Deviation	OFV	%Deviation	OFV	%Deviation	%Deviation	OFV
Rec01	20*5	1590	1541	- 3.08	1527	- 3.96	1528	- 3.90	- 3.90	1528
Rec03	20*5	1457	1433	- 1.65	1361	- 6.59	1361	- 6.59	- 6.59	1361
Rec05	20*5	1637	1590	- 2.87	1501	- 8.31	1511	- 7.70	- 7.70	1511
Rec07	20*10	2119	2083	- 1.70	2047	- 3.40	2042	- 3.59	- 3.63	2043
Rec09	20*10	2141	2138	- 0.14	2042	- 4.62	2027	- 4.58	- 5.32	2043
Rec11	20*10	1946	1954	0.41	1881	- 3.34	1881	- 2.98	- 3.34	1888
Rec13	20*15	2709	2692	- 0.63	2559	- 5.54	2545	- 6.05	- 6.05	2545
Rec15	20*15	2691	2620	- 2.64	2542	- 5.54	2529	- 6.02	- 6.02	2529
Rec17	20*15	2740	2675	- 2.37	2587	- 5.58	2587	- 5.58	- 5.58	2587
Rec19	30*10	3157	3129	- 0.89	2881	- 8.74	2861	- 9.28	- 9.38	2864
Rec21	30*10	3015	2954	- 2.02	2864	- 5.01	2822	- 5.70	- 6.40	2843
Rec23	30*10	3030	2878	- 5.02	2716	- 10.36	2700	- 10.66	- 10.89	2707
Rec25	30*15	3835	3960	3.26	3591	- 6.36	3593	- 6.23	- 6.31	3596
Rec27	30*15	3655	3690	0.96	3477	- 4.87	3431	- 6.05	- 6.13	3434
Rec29	30*15	3583	3521	- 1.73	3301	- 7.87	3291	- 8.15	- 8.15	3291
Rec31	50*10	4631	5116	10.47	4351	- 6.05	4336	- 6.37	- 6.37	4336
Rec33	50*10	4770	5274	10.57	4519	- 5.26	4466	- 5.74	- 6.37	4496
Rec35	50*10	4718	5443	15.37	4545	- 3.67	4417	- 5.87	- 6.38	4441
Rec37	75*20	8979	-	-	8361	- 6.88	8081	- 9.01	- 10.00	8170
Rec39	75*20	9158	-	-	8937	- 2.41	8517	- 6.17	- 7.00	8593
Rec41	75*20	9344	-	-	8717	- 6.71	8520	- 7.67	- 8.82	8627
		Average of %Deviation	—	0.91	-	- 5.77	-	- 6.38	- 6.68	-

Table 5 Comparison of MILP models for $FF_c|prmu|C_{max}$

Type of FFS problem	Number of binary variables	Number of continuous variables	Total number of constraints
MILP-P	$(n + 1)^2 - (n + 1)$	$n(c + 1) + 1$	$n^2(c + 1) + 5n + 2$
Samarghandi Model I	n^2	nc	$n^2(c + 1) + 4n + 1$
Wagner	$n^2(c - 2)$	$2cn + 1$	$n^3(c - 1) + 3cn$
Manne	$\frac{(c-2)n(n-1)}{2}$	$\frac{nc(n+1)}{2} + 1$	$\frac{nc(n+1)}{2}$
Stafford	n^2	$(2c - 1)n$	$n + c(n - 1) + 1$
Wilson	$n^2(c - 2)$	$nc + 1$	$n^3(c - 1) + 3nc - c + 1$
Seda	n^2	$2nc$	$nc + c + n$

Table 6 Comparison of the number of decision variables of different models

Number of Jobs (n)	50	50	50	100	100	100	200	200	200	Average	Rank	Percent deviation from the best
Number of Stations (c)	5	10	15	5	10	15	5	10	15			
<i>Number of binary variables</i>												
MILP-P	2550	2550	2550	10,100	10,100	10,100	40,200	40,200	40,200	17,617	4	1%
Samarghandi Model I	2500	2500	2500	10,000	10,000	10,000	40,000	40,000	40,000	17,500	1	0%
Wagner	7500	20,000	32,500	30,000	80,000	130,000	120,000	320,000	520,000	140,000	6	700%
Manne	3675	9800	15,925	14,850	39,600	64,350	59,700	159,200	258,700	69,533	5	297%
Stafford	2500	2500	2500	10,000	10,000	10,000	40,000	40,000	40,000	17,500	1	0%
Wilson	7500	20,000	32,500	30,000	80,000	130,000	120,000	320,000	520,000	140,000	6	700%
Seda	2500	2500	2500	10,000	10,000	10,000	40,000	40,000	40,000	17,500	1	0%
<i>Number of continuous variables</i>												
MILP-P	301	551	801	601	1101	1601	1201	2201	3201	1284	3	10%
Samarghandi Model I	250	500	750	500	1000	1500	1000	2000	3000	1167	1	0%
Wagner	501	1001	1501	1001	2001	3001	2001	4001	6001	2334	6	100%
Manne	6376	12,751	19,126	25,251	50,501	75,751	100,501	201,001	301,501	88,084	7	7450%
Stafford	450	950	1450	900	1900	2900	1800	3800	5800	2217	4	90%
Wilson	251	501	751	501	1001	1501	1001	2001	3001	1168	2	0%
Seda	500	1000	1500	1000	2000	3000	2000	4000	6000	2333	5	100%
<i>Total number of variables</i>												
MILP-P	2,851	3,101	3,351	10,701	11,201	11,701	41,401	42,401	43,401	18,901	2	1%
Samarghandi Model I	2,750	3,000	3,250	10,500	11,000	11,500	41,000	42,000	43,000	18,667	1	0%
Wagner	8,001	21,001	34,001	31,001	82,001	133,001	122,001	324,001	526,001	142,334	6	663%
Manne	10,051	22,551	35,051	40,101	90,101	140,101	160,201	360,201	560,201	157,618	7	744%
Stafford	2,950	3,450	3,950	10,900	11,900	12,900	41,800	43,800	45,800	19,717	3	6%
Wilson	7,751	20,501	33,251	30,501	81,001	131,501	121,001	322,001	523,001	141,168	5	656%
Seda	3,000	3,500	4,000	11,000	12,000	13,000	42,000	44,000	46,000	19,833	4	6%

5 Conclusions and future research directions

This study extended the CVRP model for the FFS problem which involves no-wait, time lags, and release time restrictions and the objective of minimizing the makespan. Most of the available MILP models introduced by scholars for flow shop are tailored to handle either permutation or non-permutation schedules with specific restrictions. In other words, their MILP methods cannot be applied to solve both permutation and non-permutation scheduling problems. Developing MILP formulation according to the CVRP concept would be one of the modeling approaches to remove the above limitations. The significance of this study lies in its contribution to the literature by presenting a new and efficient approach based on the CVRP problem to solve FFS scheduling problems which is the first attempt in the related

literature. Therefore, the models of the present study are versatile enough to be adapted to both types of permutation and non-permutation schedules in FFS, which can be used to optimize industrial activities. This research not only proposes multiple MILP models for various FFS problems but also presents corresponding CP models for these MILP models. Moreover, the computational performance of these parallel models is compared in various sizes.

The comparative analysis of various models reveals that CP models have superior computational performance than MILP models. However, the MILP models are also timely-efficient. Moreover, the efficiency of developed models is also represented in comparison with several benchmark datasets. Computational results from the available test problems in the literature reveal the efficiency of developed models in achieving solutions of high quality for the

Fig. 3 Number of decision variables

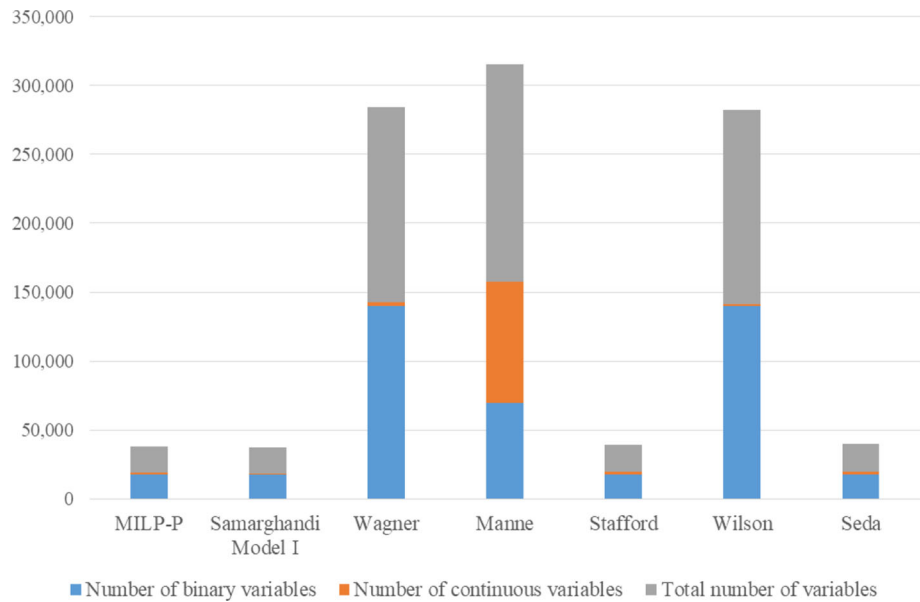


Table 7 Comparison of the number of constraints of different models

Number of Jobs (n)	50	50	50	100	100	100	200	200	200	Average	Rank
Number of Stations (c)	5	10	15	5	10	15	5	10	15		
Stafford	296	541	786	596	1,091	1,586	1,196	2,191	3,186	1,274	1
Seda	305	560	815	605	1,110	1,615	1,205	2,210	3,215	1,293	2
Manne	6,375	12,750	19,125	25,250	50,500	75,750	100,500	201,000	301,500	88,083	3
Samarghandi Model I	15,201	27,701	40,201	60,401	110,401	160,401	240,801	440,801	640,801	192,968	4
MILP-P	15,252	27,752	40,252	60,502	110,502	160,502	241,002	441,002	641,002	193,085	5
Wilson	500,746	1,126,491	1,752,236	4,001,496	9,002,991	14,004,486	32,002,996	72,005,991	112,008,986	27,378,491	6
Wagner	500,750	1,126,500	1,752,250	4,001,500	9,003,000	14,004,500	32,003,000	72,006,000	112,009,000	27,378,500	7

given problems in a reasonable time. The extended models can improve many solutions of the small and large best-known test problems. Based on the findings, while the objective function values of the MILP and CP models in non-permutation schedules exhibit lower values than their respective equivalents in permutation schedules, both models demonstrate longer runtime in non-permutation schedules. Therefore, enterprises should assess the computational performance of various models and select the most appropriate one (case-by-case) based on the problem size and complexity. The extended models can be utilized for a wide range of real-world FFS scheduling problems in manufacturing processes by inputting the related parameters into the MILP or CP model. The models can also assist the enterprise to evaluate various production scenarios (like changing the number of machines) and compare their impact on the makespan. This

research provides significant insights for decision-makers and managers in industries that require efficient scheduling methods. However, a limitation of the research is the lack of a comprehensive dataset in the literature that considers all the restrictions in permutation and non-permutation schedules.

Future studies may focus on applying these proposed models to various scheduling problems like the blocking flow shop scheduling problem or the problems with other objectives or restrictions, like the precedence relations. The flow shop models presented in the current study are categorized into identical Parallel Machine Scheduling (PMS) problems. A future study could be conducted by extending the modeling approach proposed in this research to solve uniform and unrelated PMS problems. Moreover, other time performance criteria such as flow time, maximum tardiness and mean tardiness (or earliness) could be taken into account instead of the

makespan or as a new objective function. It would be interesting to extend this research to other manufacturing settings like job shops and open shops.

References

- Boufellouh, R., Belkaid, F.: Bi-objective optimization algorithms for joint production and maintenance scheduling under a global resource constraint: application to the permutation flow shop problem. *Comput. Oper. Res.* **122**, 104943 (2020)
- Legarretaetxebarria, A., Quartulli, M., Olaizola, I., Serrano, M.: Optimal scheduling of manufacturing processes across multiple production lines by polynomial optimization and bagged bounded binary knapsack. *Int. J. Interact. Des. Manuf. (IJDeM)* **11**(1), 83–91 (2017)
- Bagchi, T.P., Gupta, J.N., Sriskandarajah, C.: A review of TSP based approaches for flowshop scheduling. *Eur. J. Oper. Res.* **169**(3), 816–854 (2006)
- Öztop, H., Tasgetiren, M.F., Eliyi, D.T., Pan, Q.-K.: Metaheuristic algorithms for the hybrid flowshop scheduling problem. *Comput. Oper. Res.* **111**, 177–196 (2019)
- Lei, C., Zhao, N., Ye, S., Wu, X.: Memetic algorithm for solving flexible flow-shop scheduling problems with dynamic transport waiting times. *Comput. Ind. Eng.* **139**, 105984 (2020)
- Almasarwah, N., Süer, G.A.: Consideration of processing time dissimilarity in batch-cyclic scheduling of flowshop cells. *Int. J. Prod. Res.* **59**(21), 6544–6563 (2020)
- Shabtay, D., Arviv, K.: Optimal robot scheduling to minimize the makespan in a three-machine flow-shop environment with job-independent processing times. *Appl. Math. Model.* **40**(5), 4231–4247 (2016)
- Mucha, M., Sviridenko, M.: No-wait flowshop scheduling is as hard as asymmetric traveling salesman problem. *Math. Oper. Res.* **41**(1), 247–254 (2016)
- Dorfeshan, Y., Tavakkoli-Moghaddam, R., Mousavi, S.M., Vahedi-Nouri, B.: A new weighted distance-based approximation methodology for flow shop scheduling group decisions under the interval-valued fuzzy processing time. *Appl. Soft Comput.* **91**, 106248 (2020)
- Bautista-Valhondo, J., Alfaro-Pozo, R.: Mixed integer linear programming models for Flow Shop Scheduling with a demand plan of job types. *CEJOR* **28**(1), 5–23 (2020)
- Ozolins, A.: Improved bounded dynamic programming algorithm for solving the blocking flow shop problem. *CEJOR* **27**(1), 15–38 (2019)
- Ćwik, M., Józefczyk, J.: Heuristic algorithms for the minmax regret flow-shop problem with interval processing times. *CEJOR* **26**(1), 215–238 (2018)
- Amrouche, K., Boudhar, M., Sami, N.: Two-machine chain-reentrant flow shop with the no-wait constraint. *Eur. J. Ind. Eng.* **14**(4), 573–597 (2020)
- Li, X., Chan, F.T., Chung, S.: Optimal multi-degree cyclic scheduling of multiple robots without overlapping in robotic flowshops with parallel machines. *J. Manuf. Syst.* **36**, 62–75 (2015)
- Dhouib, E., Teghem, J., Loukil, T.: Lexicographic optimization of a permutation flow shop scheduling problem with time lag constraints. *Int. Trans. Oper. Res.* **20**(2), 213–232 (2013)
- Meziani, N., Boudhar, M., Oulamara, A.: PSO and simulated annealing for the two-machine flowshop scheduling problem with coupled-operations. *Eur. J. Ind. Eng.* **12**(1), 43–66 (2018)
- Mkadem, M.A., Moukrim, A., Serairi, M.: Exact method for the two-machine flow-shop problem with time delays. *Ann. Oper. Res.* **298**(1), 375–406 (2021)
- Arora, D., Agarwal, G.: Meta-heuristic approaches for flowshop scheduling problems: a review. *Int. J. Adv. Oper. Manag.* **8**(1), 1–16 (2016)
- Wang, G., Li, X., Gao, L., Li, P.: An effective multi-objective whale swarm algorithm for energy-efficient scheduling of distributed welding flow shop. *Annals Oper. Res.* **310**, 223–255 (2021)
- Błażewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. (2007). *Handbook on scheduling: from theory to applications*: Springer Science & Business Media.
- Emmons, H., & Vairaktarakis, G. (2012). *Flow shop scheduling: theoretical results, algorithms, and applications*: Springer Science & Business Media (Vol. 182)
- Yimer, A.D., Demirli, K.: Fuzzy scheduling of job orders in a two-stage flowshop with batch-processing machines. *Int. J. Approx. Reason.* **50**(1), 117–137 (2009). <https://doi.org/10.1016/j.ijar.2007.08.013>
- Gupta, J.N., Stafford, E.F.: Flowshop scheduling research after five decades. *Eur. J. Oper. Res.* **169**(3), 699–711 (2006)
- Booth, K. E., Tran, T. T., Nejat, G., & Beck, J. C. (2016). Mixed-integer and constraint programming techniques for mobile robot task planning.
- IBM. (2012). *IBM ILOG CPLEX Optimization Studio 12.5 User's Manual*.
- Li, H., Womer, K.: Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. *J. Sched.* **12**(3), 281 (2009)
- Rossi, F., Van Beek, P., & Walsh, T. (2006). *Handbook of constraint programming*. Elsevier.
- Novas, J.M., Henning, G.P.: A comprehensive constraint programming approach for the rolling horizon-based scheduling of automated wet-etch stations. *Comput. Chem. Eng.* **42**, 189–205 (2012)
- Rossit, D.A., Vásquez, Ó.C., Tohmé, F., Frutos, M., Safe, M.D.: A combinatorial analysis of the permutation and non-permutation flow shop scheduling problems. *Eur. J. Oper. Res.* **289**(3), 841–854 (2021)
- Singh, H., Oberoi, J.S., Singh, D.: Multi-objective permutation and non-permutation flow shop scheduling problems with no-wait: a systematic literature review. *RAIRO-Oper. Res.* **55**(1), 27–50 (2021)
- Samarghandi, H.: Minimizing the makespan in a flow shop environment under minimum and maximum time-lag constraints. *Comput. Ind. Eng.* **136**, 614–634 (2019)
- Qin, T., Du, Y., Chen, J.H., Sha, M.: Combining mixed integer programming and constraint programming to solve the integrated scheduling problem of container handling operations of a single vessel. *Eur. J. Oper. Res.* **285**(3), 884–901 (2020)
- Xiao, Y., Yuan, Y., Zhang, R.Q., Konak, A.: Non-permutation flow shop scheduling with order acceptance and weighted tardiness. *Appl. Math. Comput.* **270**, 312–333 (2015)
- Xin, X., Jiang, Q., Li, S., Gong, S., Chen, K.: Energy-efficient scheduling for a permutation flow shop with variable transportation time using an improved discrete whale swarm optimization. *J. Clean. Prod.* **293**, 126121 (2021)
- Méndez, C., Henning, G., Cerdá, J.: An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Comput. Chem. Eng.* **25**(4), 701–711 (2001)
- Fondrevelle, J., Oulamara, A., Portmann, M.-C.: Permutation flowshop scheduling problems with maximal and minimal time lags. *Comput. Oper. Res.* **33**(6), 1540–1556 (2006)
- Liao, C.-J., Liao, L.-M.: Improved MILP models for two-machine flowshop with batch processing machines. *Math. Comput. Model.* **48**(7), 1254–1264 (2008)
- Fondrevelle, J., Oulamara, A., Portmann, M.-C.: Permutation flowshop scheduling problems with time lags to minimize the weighted

- sum of machine completion times. *Int. J. Prod. Econ.* **112**(1), 168–176 (2008)
39. Ruiz, R., Şerifoğlu, F.S., Urlings, T.: Modeling realistic hybrid flexible flowshop scheduling problems. *Comput. Oper. Res.* **35**(4), 1151–1175 (2008)
 40. Gong, H., Tang, L., Duin, C.: A two-stage flow shop scheduling problem on a batching machine and a discrete machine with blocking and shared setup times. *Comput. Oper. Res.* **37**(5), 960–969 (2010)
 41. Gicquel, C., Hege, L., Minoux, M., Van Canneyt, W.: A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints. *Comput. Oper. Res.* **39**(3), 629–636 (2012)
 42. Zhou, Z., Che, A., Yan, P.: A mixed integer programming approach for multi-cyclic robotic flowshop scheduling with time window constraints. *Appl. Math. Model.* **36**(8), 3621–3629 (2012)
 43. Costa, A., Cappadonna, F.A., Fichera, S.: A dual encoding-based meta-heuristic algorithm for solving a constrained hybrid flow shop scheduling problem. *Comput. Ind. Eng.* **64**(4), 937–958 (2013)
 44. Fumero, Y., Corsano, G., Montagna, J.M.: A Mixed Integer Linear Programming model for simultaneous design and scheduling of flowshop plants. *Appl. Math. Model.* **37**(4), 1652–1664 (2013)
 45. Arabameri, S., Salmasi, N.: Minimization of weighted earliness and tardiness for no-wait sequence-dependent setup times flowshop scheduling problem. *Comput. Ind. Eng.* **64**(4), 902–916 (2013)
 46. Lei, W., Che, A., Chu, C.: Optimal cyclic scheduling of a robotic flowshop with multiple part types and flexible processing times. *Eur. J. Ind. Eng.* **8**(2), 143–167 (2014)
 47. Soltani, S.A., Karimi, B.: Cyclic hybrid flow shop scheduling problem with limited buffers and machine eligibility constraints. *Int. J. Adv. Manufact. Technol.* **76**(9–12), 1739–1755 (2015)
 48. Xiong, F., Xing, K., Wang, F.: Scheduling a hybrid assembly-differentiation flowshop to minimize total flow time. *Eur. J. Oper. Res.* **240**(2), 338–354 (2015)
 49. Zabihzadeh, S.S., Rezaeian, J.: Two meta-heuristic algorithms for flexible flow shop scheduling problem with robotic transportation and release time. *Appl. Soft Comput.* **40**, 319–330 (2016)
 50. Samarghandi, H., Behroozi, M.: On the exact solution of the no-wait flow shop problem with due date constraints. *Comput. Oper. Res.* **81**, 141–159 (2017). <https://doi.org/10.1016/j.cor.2016.12.013>
 51. Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.R.: Optimization and approximation in deterministic sequencing and scheduling: a survey, pp. 287–326. Elsevier, In *Annals of discrete mathematics* (1979)
 52. Hillier, F.S., Price, C.C.: *International Series in Operations Research & Management Science*. Springer, Berlin/Heidelberg, Germany (2005)
 53. Milano, M., Trick, M.: *Constraint and integer programming Constraint and Integer Programming*, pp. 1–31. Springer, Cham (2004)
 54. Gedik, R., Kalathia, D., Egilmez, G., Kirac, E.: A constraint programming approach for solving unrelated parallel machine scheduling problem. *Comput. Ind. Eng.* **121**, 139–149 (2018)
 55. Gökgür, B., Hnich, B., Özpeynirci, S.: Parallel machine scheduling with tool loading: a constraint programming approach. *Int. J. Prod. Res.* **56**(16), 5541–5557 (2018)
 56. Rocha, P.L., Ravetti, M.G., Mateus, G.R., Pardalos, P.M.: Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. *Comput. Oper. Res.* **35**(4), 1250–1264 (2008)
 57. Carlier, J.: Ordonnancements a contraintes disjonctives *Revue française d'automatique d'informatique et de recherche opérationnelle. Recherche opérationnelle* **12**(4), 333–350 (1978)
 58. Taillard, E.: Some efficient heuristic methods for the flow shop sequencing problem. *Eur. J. Oper. Res.* **47**(1), 65–74 (1990)
 59. Manne, A.S.: On the job-shop scheduling problem. *Oper. Res.* **8**(2), 219–223 (1960)
 60. Rajendran, C.: A no-wait flowshop scheduling heuristic to minimize makespan. *J. Oper. Res. Soc.* **45**(4), 472–478 (1994)
 61. Samarghandi, H.: A particle swarm optimisation for the no-wait flow shop problem with due date constraints. *Int. J. Prod. Res.* **53**(9), 2853–2870 (2015)
 62. Samarghandi, H., ElMekkawy, T.Y.: A meta-heuristic approach for solving the no-wait flow-shop problem. *Int. J. Prod. Res.* **50**(24), 7313–7326 (2012)
 63. Wilson, J.: Alternative formulations of a flow-shop scheduling problem. *J. Opera. Res. Soc.* **40**(4), 395–399 (1989)
 64. Wagner, H.M.: An integer linear-programming model for machine scheduling. *Naval Res. Logist. (NRL)* **6**(2), 131–140 (1959)
 65. Pan, C.-H.: A study of integer programming formulations for scheduling problems. *Int. J. Syst. Sci.* **28**(1), 33–41 (1997)
 66. Stafford, E.F.: On the development of a mixed-integer linear programming model for the flowshop sequencing problem. *J. Oper. Res. Soc.* **39**(12), 1163–1174 (1988)
 67. Šeda, M.: Mathematical models of flow shop and job shop scheduling problems. *World Acad. Sci. Eng. Technol.* **1**(31), 122–127 (2007)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.