Applied Mathematical Modelling 37 (2013) 977–988

Contents lists available at SciVerse ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Evaluation of mathematical models for flexible job-shop scheduling problems

Yunus Demir *, S. Kürşat İşleyen

Faculty of Engineering, Department of Industrial Engineering, Atatürk University, 25240 Erzurum, Turkey

ARTICLE INFO

Article history:
Received 16 August 2011
Received in revised form 7 March 2012
Accepted 12 March 2012
Available online 21 March 2012

Keywords:
Flexible job-shops
Scheduling
Mixed integer programming

ABSTRACT

With the rapid development in computer technologies, mathematical programming-based technique to solve scheduling problems is significantly receiving attention from researchers. Although, it is not efficient solution method due to the NP-hard structure of these problems, mathematical programming formulation is the first step to develop an effective heuristic. Numerous comparative studies for variety scheduling problems have appeared over the years. But in our search in literature there is not an entirely review for mathematical formulations of flexible job shop scheduling problems (FJSP). In this paper, four the most widely used formulations of the FJSP are compiled from literature and a time-indexed model for FJSP is proposed. These formulations are evaluated under three categories that are distinguished by the type of binary variable that they rely on for using of sequencing operations on machines. All five formulations compared and results are presented.

1. Introduction

Scheduling can be defined as the allocation of resources to perform a collection of tasks over a period of time. Finding the best schedule can be very easy or very difficult, depending on the shop environment, the process constraints and the performance indicator [1]. One of the most difficult problems in this area is the job shop scheduling problem (JSP), where a set of n jobs must be processed on m machines where each job i consists of n_i operations that should be performed on the machines while satisfying precedence constraints. JSP aims to find the appropriate sequencing of operations on the machines to optimize the performance indicator. A typical performance indicator for JSP is the makespan, i.e., the time needed to complete all jobs. It is well known that this problem is NP-hard [2].

The FJSP is an extension of the classical JSP, where operations are allowed to be processed on any among a set of available machines [3]. This makes FJSP more difficult to solve due to the consideration of both routing of jobs and scheduling of operations. Therefore FJSP is NP-hard too.

As for the mathematical models the initial formulations for scheduling problems were devised starting around 1960 [4]. Wagner [5], Bowman [6], Manne [7] presented three distinct ways of formulating the sequencing problem using integer programming (IP). These three approaches are distinguished by the type of binary variable that captures the sequencing decision. Other formulations in literature tend to be hybrid combinations of these three or elaborations based on detailed computational considerations [8].

Although mathematical programming formulation is not efficient solution method due to the NP-hard structure of machine scheduling problems, it is first key step prior to developing an effective heuristics and useful to understand the structure of the problem [9]. Hence, researchers in this field should be aware of the relative efficiency of scheduling models [10].

* Corresponding author. Tel.: +90 442 2314722.
E-mail addresses: demiry@atauni.edu.tr (Y. Demir), kursat.isleyen@atauni.edu.tr (S. Kürşat İşleyen).

0307-904X/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.apm.2012.03.020

Numerous machine scheduling comparative studies have appeared in literature. Blazewichz et al. [11] is the first one that focuses on mathematical models for scheduling problems. They presented mathematical programming formulations for single-machine, parallel-machine and job shop scheduling problems. Pan [10] has compared mathematical models for job-shop and flow-shop scheduling problems described in the literature in his review paper. Keha et al. [12] have provided a review and a comparison of mixed integer linear programming (MILP) formulations for single machine scheduling problems. Unlu and Mason [9] presented four different mixed integer programming (MIP) formulations based on four different types of decision variables for various parallel machine scheduling problems. Pan and Chen [13] described the development of mixed binary integer programming formulations for the reentrant job shop scheduling problem in their paper.

As presented above various survey papers have appeared on mathematical programming formulations for machine scheduling problems over the years but in our search none of them except one focused on mathematical formulations for FJSPs. The exception is paper presented by Özgüven et al. [4]. They developed a mixed integer linear programming model for FJSP and compared to a model of Fattahi et al. [14] in terms of computational efficiency.

In this paper, articles including mathematical models related FJSP are investigated in terms of binary variables that they rely on for using of sequencing operations on machines named sequence-position variable, precedence variable and time indexed variable. By assuming the objective function is makespan computational efficiency of models is compared and an overview of models formulated for FJSP in the literature is given in Table 1.

The remainder of this paper is organized as follows: in Section 2, problem definition of FJSP and notation of models is presented. In Section 3, Classification paradigm is described. In Section 4, mathematical formulations for FJSP are presented. In Section 5, computational results of models compared. Finally the conclusions of study are drawn in section 6.

2. Problem definition and notations

This problem has m machine and n jobs. Each job consists of a sequence of operation where they are allowed to be processed on any among a set of available machines. All jobs and machines are available at time 0, and a machine can only execute one operation at a given time. Preemption is not allowed. Models evaluated in this paper setting up times of machines and transportation times between operations are neglected and objective is considered to minimize the Cmax. The notation describes the indices, parameters, and decision variables used in the models are as follows (for each model all notations are not required):

Indices
i, h: index of jobs $(1,\ldots,n)$
j, g: index of operations $(1,\ldots,J_i)$
k: index of machines $(1,\ldots,m)$
l: sequence of assigned operation on machine k $(1,\ldots d_k)$
u: index of time period
Parameters
n: total number of jobs
m: total number of machines
J_i: total number of operations of job i
a_{kij}: Describe the capable machines set M_{ij} is assigned to operation O_{ij}

$$a_{kij} : \begin{cases} 1, & \text{if } O_{ij} \text{ can be performed on machine } i \\ 0, & \text{otherwise} \end{cases}$$

p_{kij}: processing time of O_{ij} if performed on machine k
M: a large number
E_k: the set of operations which can be performed on machine k
Decision variables
Cmax: makespan
c_{ij}: completion time of operation O_{ij}
s_{ijk}: starting time of operation O_{ij} on machine k
c_{ijk}: completion time of operation O_{ij} on machine k
c_i: completion time of job i

$$x_{ijkl} : \begin{cases} 1, & \text{if } O_{ij} \text{ is performed on machine } k \text{ in priority } l \\ 0, & \text{otherwise} \end{cases}$$

$$v_{ijk} : \begin{cases} 1, & \text{if } O_{ij} \text{ performed on machine } k \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ijhgk}: \begin{cases} 1, & \text{if operation } O_{ij} \text{ precedes operation } O_{hg} \text{ on machine } k \\ 0, & \text{otherwise} \end{cases}$$

$$w_{ijku}: \begin{cases} 1, & \text{if } O_{ij} \text{ is processed by machine during period } u \\ 0, & \text{otherwise} \end{cases}$$

t_{ij}: starting time of operation O_{ij}
Tm_{kl}: start of working time for machine k in priority l
d_k: the number of assigned operations to machine k
ps_{ij}: Processing time of operation O_{ij} after select a machine

3. A preliminary study

Unlike the classical JSP where each operation is processed on a predefined machine, each operation in the FJSP can be processed on one out of several machines. The problem of scheduling jobs in FJSP could be decomposed into two sub-problems: the routing sub-problem that assigns each operation to a machine selected out of a set of capable machines, the scheduling sub-problem that consists of sequencing the assigned operations on all machines in order to obtain a feasible schedule to minimize the predefined objective function [15].

Review of the literature reveals that for routing sub-problem v_{ijk} binary variable is used which equals 1, if operation O_{ij} performed on machine k otherwise 0. For sequencing sub-problem there exist the following three different definitions of binary variables:

3.1. Sequence-position variable

In this type, we have the sequence-position variable, proposed by Wagner [5]. These variables are defined based on the notion that each machine has a fixed number of positions or slots into which jobs can be assigned. These positions by construction specify a job's relative position to all other jobs processed on the same machine and therefore, the job sequence on the machine. Let binary variables $x_{ijkl} = 1$ if operation O_{ij} is assigned to position l on machine k; otherwise, $x_{ijkl} = 0$ [9].

3.2. Precedence variable

This approach relies on precedence variable z_{ijhgk}, introduced by Manne [7]. It denotes the sequence of operations assigned on same machine. It is equal to one if operation O_{ij} precedes operation O_{hg} on machine k otherwise zero. Note that operation O_{ij} is not necessarily positioned immediately before operation O_{hg} when $z_{ijhgk} = 1$ [9]. For this type of variable it has to be defined only $i < h$ because $z_{ijhgk} = 1 - z_{hgijk}$ and z_{ijijk} or z_{hghgk} are not needed. Presented models below precedence variables are used in disjunctive constraints, meaning that one or the other must hold for solution to be feasible [4].

3.3. Time-indexed variable

This approach is based on time indexed variables proposed by Bowman [6]. Let the binary variable $w_{ijku} = 1$ if operation O_{ij} processes in period u on machine k and is equal to zero otherwise. In the time indexed model planning horizon is considered as discrete.

4. Mathematical formulations

There existed lots of mathematical model formulated for FJSP with kinds of objectives (Minimum Cmax, total tardiness etc...) and constraints (Set up time, buffer size etc.) (Table 1). Mathematical models that will be presented in this section are taken from mostly used ones from the literature and examined under three groups distinguished by their binary variables that are used for sequencing operations on machines. For all models below to indicate assigning operation on machines, variable v_{ijk} is used which equals 1 if operation O_{ij} is assigned on machine k, otherwise 0.

4.1. Sequence-position variable based model

This type of variable is first proposed by Wagner [5] to formulate JSP. As for FJSP it was first used by Lee et al. [20]. Fattahi et al. used this modeling technique to formulate FJSP [14], FJSP with overlapping consideration [33], multi objective FJSP with overlapping consideration [34] and dynamic scheduling in FJSP [19]. Also this modeling technic is used to formulate scheduling problems in virtual manufacturing cells which is different from FJSP an operation of job can be processed simultaneously different alternative machines and distances between machines are considered [36,37].

Table 1
The articles including mathematical models for FJSP.

References	Comments	Objective	Model type*
[16]	FJSP with transportation constraints and bounded processing time	Min. (Cmax and storage)	2
[17]	Multi-objective FJSP	Min. (Cmax, max. machine workload, total machine workload)	2
[18]	Bi-objective FJSP with preventive maintenance	Min. (Cmax, system unavailability for the maintenance part)	2
[19]	Dynamic scheduling in FJSP	Min. (efficiency and stability)	1
[20]	FJSP with outsourcing, due date and with process plan flexibility	Min. (Cmax)	1
[21]	FJSP with sequence dependent setup times	Min. (Cmax)	2
[22]	FJSP	Min. (total weighted quadratic tardiness)	3
[14]	FJSP	Min. (Cmax)	1
[23]	FJSP with sequence independent setup times	Min. (total tardiness)	2
[4]	FJSP	Min. (Cmax)	2
[4]	FJSP – with process plan flexibility	Min. (Cmax)	2
[24]	FJSP with sequence dependent setup times	Min. (Cmax)	2
[25]	FJSP with following time and sequence independent setup times	Min. (Cmax)	2
[26]	Multi-objective FJSP with sequence independent setup times	Min. (mean job flow time, mean job tardiness, and minimum mean machine idle time)	2
[27]	Multi-objective FJSP in consideration of maintenance, sequence dependent setup times and intermediate buffer	Min. (Cmax, idleness of machines and interruption)	2
[28]	Multi-objective FJSP	Min. (Cmax, max. machine workload, total machine workload)	2
[29]	FJSP with identical machines and consideration of overlapping and buffer	Min. (costs of in-process inventory, orders not fully completed at the end of the scheduling horizon and cost derived from failing to meet the 'just in time' due dates)	3
[30]	FJSP	Min. (Cmax)	2
[31]	Multi-objective FJSP with non-fixed availability constraints	Min. (Cmax, max machine workload, total machine workload)	2
[32]	FJSP with identical machines	Min. (sum of setup and inventory holding costs per time unit)	1
[33]	FJSP with overlapping consideration	Min. (Cmax)	1
[34]	Multi-objective FJSP with overlapping consideration	Min. (Cmax, critical machine work loading and total work loading time)	1
[35]	Multi-objective FJSP with sequence dependent setup times, storage and transportation constraints	Min. (Cmax, mean completion time, max tardiness)	2

* Model types based on binary variable: (1) sequence-position variable based model, (2) precedence variable based model, (3) time indexed model

4.1.1. Model M1

$$\mathrm{Cmax} \geqslant t_{ij} + ps_{ij} \quad \forall i, j = J_i \tag{1.1}$$

$$\Sigma_k p_{kij} * V_{ijk} = ps_{ij} \quad \forall i, j \tag{1.2}$$

$$t_{ij} + ps_{ij} \leqslant t_{ij+1} \quad \forall i, \quad \forall j = 1, \ldots, J_i - 1 \tag{1.3}$$

$$Tm_{kl} + ps_{ij} * x_{ijkl} \leqslant Tm_{kl+1} \quad \forall i, j, k, \quad \forall l = 1, \ldots, d_k - 1 \tag{1.4}$$

$$Tm_{kl} \leqslant t_{ij} + (1 - x_{ijkl}) * M \quad \forall i, j, k, l \tag{1.5}$$

$$Tm_{kl} + (1 - x_{ijkl}) * M \geqslant t_{ij} \quad \forall i, j, k, l \tag{1.6}$$

$$V_{ijk} \leqslant a_{kij} \quad \forall i, j, k \tag{1.7}$$

$$\Sigma_i \Sigma_j x_{ijkl} = 1 \quad \forall k, l \tag{1.8}$$

$$\Sigma_k v_{ijk} = 1 \quad \forall i, j \tag{1.9}$$

$$\Sigma_l x_{ijkl} = v_{ijk} \quad \forall i, j, k \tag{1.10}$$

$$t_{ij} \geqslant 0 \quad \forall i, j \tag{1.11}$$

$$ps_{ij} \geqslant 0 \quad \forall i, j \tag{1.12}$$

$$Tm_{kl} \geqslant 0 \quad \forall k, l \tag{1.13}$$

$$x_{ijkl} \in \{0, 1\} \quad \forall i, j, k, l \tag{1.14}$$

$$v_{ijk} \in \{0, 1\} \quad \forall i, j, k \tag{1.15}$$

Constraint (1.1) determines the makespan. Constraint (1.2) determines the processing time of operation O_{ij} by selected machine. Constraint (1.3) describes the operation precedence constraints. Constraint (1.4) forces each machine to process one operation at a time. Constraints (1.5 and 1.6) force each operation O_{ij} can be start after its assigned machine is idle and previous operation O_{ij} is completed. Constraint (1.7) determines the capable machines for each operation. Constraint (1.8) assigns the operations to a machine and sequence assigned operations on all machines. Constraints (1.9) and (1.10) force each operation can be performed only on one machine and at one priority.

4.2. Precedence variable based model

This approach introduced by Manne [7]. We reviewed the literature and mostly used three model types in this approach for FJSP presented below:

4.2.1. Model M2

This kind of model proposed by Özgüven et al. [4] to formulate FJSP and FJSP with process plan flexibility.

$$\text{Cmax} \geqslant c_i \quad \forall i \tag{2.1}$$

$$c_i \geqslant \Sigma_{k \in Mij} c_{ijk} \quad \forall i, j = J_i \tag{2.2}$$

$$s_{ijk} + c_{ijk} \leqslant v_{ijk} * M \quad \forall i, j, \forall k \in Mij \tag{2.3}$$

$$c_{ijk} \geqslant s_{ijk} + p_{kij} - (1 - v_{ijk}) * M \quad \forall i, j, \forall k \in Mij \tag{2.4}$$

$$s_{ijk} \geqslant c_{hgk} - (z_{ijhgk}) * M \quad \forall i \leqslant h, \forall j, g, \forall k \in Mij \cap Mhg \tag{2.5}$$

$$S_{hgk} \geqslant c_{ijk} - (1 - z_{ijhgk}) * M \quad \forall i \leqslant h, \forall j, g, \forall k \in Mij \cap Mhg \tag{2.6}$$

$$\Sigma_{k \in Mij} s_{ijk} \geqslant \Sigma_{k \in Mij} c_{ij-1k} \quad \forall i, \forall j = 2, \ldots, Ji \tag{2.7}$$

$$\Sigma_{k \in Mij} v_{ijk} = 1 \quad \forall i, j \tag{2.8}$$

$$s_{ijk}, \geqslant 0, c_{ijk} \geqslant 0 \quad \forall i, j, k \tag{2.9}$$

$$c_i \geqslant 0 \quad \forall i \tag{2.10}$$

$$v_{ijk} \in \{0, 1\} \forall i, j, k \tag{2.11}$$

$$z_{ijhgk} \in \{0, 1\} \quad \forall i \leqslant h, \forall j, g, \forall k \in Mij \cap Mhg \tag{2.12}$$

Constraint (2.1) determines the makespan. Constraint (2.2) determine the completion times (of the final operations) of the jobs. Constraint (2.4) guarantee that the difference between the starting and the completion times is equal in the least to the processing time on machine k. Constraints (2.5) and (2.6) take care of the requirement that operation O_{ij} and operation O_{hg} cannot be done at the same time on any machine in the set $Mij \cap Mhg$. Constraint (2.7) ensures that the precedence relationships between the operations of a job are not violated, i.e. the operation O_{ij} is not started before the operation O_{ij-1} has been completed. A constraint (2.8) ensures that an operation is performed on one and only one machine. If operation O_{ij} is not assigned to machine k, Constraint (2.3) set the starting and completion times of it on machine k equal to zero if operation O_{ij} is not assigned to machine k.

4.2.2. Model M3

Model presented below is constructed based on completion time of operations c_{ij} and precedence variable z_{ijhgk}. Formulated for FJSP first by Gao et al. [31] with preventive maintenance task. Then Imanipour [24] used this kind of formulation for FJSP with sequence dependent set up time, Gen et al. [28] and Zhang et al. [17] for multi-objective FJSP. They solved this problem for same objectives but with different heuristics. Finally Moradi et al. [18] used for bi-objective FJSP with preventive maintenance.

$$\text{Cmax} \geqslant c_{ij} \forall i, j = J_i \tag{3.1}$$

$$c_{ij} - c_{ij-1} \geqslant p_{kij} * v_{ijk} \quad \forall i, k, \forall j = 2, \ldots, J_i \tag{3.2}$$

$$c_{ij} \geqslant p_{kij} * v_{ijk} \quad \forall i, j = 1, k \in Mij \tag{3.3}$$

$$(c_{hg} - c_{ij} - p_{khg}) * v_{hgk} * v_{ijk} * z_{ijhgk} \geqslant 0 \quad \forall i, h, j, g, k \in Mij \cap Mhg \tag{3.4}$$

$$(c_{ij} - c_{hg} - p_{kij}) * v_{ijk} * v_{hgk} * z_{hgijk} \geqslant 0 \quad \forall i, h, j, g, k \in Mij \cap Mhg \tag{3.5}$$

$$\Sigma_{k \in Mij} v_{ijk} = 1 \quad \forall i, j \tag{3.6}$$

$$z_{ijhgk} + z_{hgijk} = v_{ijk} * v_{hgk} \quad \forall i, h, j, g, k \in Mij \cap Mhg \tag{3.7}$$

$$c_{ij} \geqslant 0 \quad \forall i, j \tag{3.8}$$

$$v_{ijk} \in \{0, 1\} \quad \forall i, j, k \tag{3.9}$$

Constraint (3.1) determines the makespan. Constraint (3.2) enforces each job to follow a specified operation sequence. Constraint (3.3) ensures the completion time of first operation of job i equal to be at least the processing time of O_{ij}. Constraints (3.4 and 3.5) are disjunctive constraints. It represents that the operation O_{hg} should be not be started before the completion of operation O_{ij}, or that the operation O_{hg} must be completed before the starting of operation O_{ij} if they are assigned on the same machine k. Constraint (3.6) states that one machine must be selected from a set of available machines for each operation. Constraint (3.7) enforces to be chosen one of two precedence relationships.

4.2.3. Model M4

Model below formulated based on variable s_{ijk} starting time of operation on assigned machine k and precedence variable. For FJSP it was used by Kim and Egbelu [30] first. Then different kind of versions of FJSP formulated. Low and Wu [23] used this formulation with sequence independent set up times, Low et al. [26] used for multi-objective FJSP, Mehrabad and Fattahi [21] used with sequence dependent set up time and Zhang et al. [16] used with transportation constraints and bounded processing time.

$$Cmax \geqslant t_{ij} + \Sigma_k p_{kij} * v_{ijk} \quad \forall i, j \tag{4.1}$$

$$t_{ij} + \Sigma_k p_{kij} * v_{ijk} \leqslant t_{ij+1} \quad \forall i, \forall j = 1, \ldots, Ji - 1 \tag{4.2}$$

$$\Sigma_{k \in Mij} v_{ijk} = 1 \quad \forall i, j \tag{4.3}$$

$$s_{ijk} = t_{ij} * v_{ijk} \quad \forall i, j, k \tag{4.4}$$

$$s_{ijk} + v_{ijk} * p_{kij} - M * (1 - z_{ijhgk}) \leqslant s_{hgk} \quad \forall i, j, h, g \in E_k, O_{ij} \neq O_{hg} \tag{4.5}$$

$$s_{hgk} + v_{hgk} * p_{khg} - M * (1 - z_{hgijk}) \leqslant s_{ijk} \quad \forall i, j, h, g \in E_k, O_{ij} \neq O_{hg} \tag{4.6}$$

$$z_{ijhgk} + z_{hgijk} = v_{ijk} * v_{hgk} \quad \forall i, j, h, g \in E_k, O_{ij} \neq O_{hg} \tag{4.7}$$

$$t_{ij} \geqslant 0 \quad \forall i, j \tag{4.8}$$

$$s_{ijk} \geqslant 0 \quad \forall i, j, k \tag{4.9}$$

$$z_{ijhgk}, z_{hgijk} \in \{0 - 1\} \quad \forall i, j, h, g \in E_k, \quad O_{ij} \neq O_{hg} \tag{4.10}$$

$$v_{ijk}, v_{hgk} \in \{0 - 1\} \quad \forall i, j, h, g \tag{4.11}$$

Constraint (4.1) determines the makespan. Constraint (4.2) presents the precedence relationship. Constraint (4.3) makes sure that operation O_{ij} is assigned to only one machine. if O_{ij} is processed on machine k, Constraint (4.4) makes start of working time of machine k for operation O_{ij} (s_{ijk}) is equal to starting time of O_{ij} (t_{ij}). If both (O_{ij},O_{hg}) operations are processed on machine k, Constraints (4.5 and 4.6) conclude the corresponding orientation of a possible operation pair and also ensure that each machine can process only one job at a time. If both operations are processed on the same machine k, Constraint (4.7) determines whether the disjunctive arcs of each possible operation pair (O_{ij},O_{hg}) exist.

4.3. Time-indexed model

Time indexed variables, which assign operations to time periods of capable machine indicated w_{ijku} which equals 1, if O_{ij} is processed by machine k during period u otherwise 0. Thomalla [22] and Gomes et al. [29] formulated model based on time-indexed variable for FJSP with parallel (identical) machines. In our search in the literature for time-indexed model for FJSP with unparallel machines (i.e., capable machines with possibly different efficiency for an operations), we did not come across. The model below we proposed to be an example for time-indexed model for FJSP with unparallel machines.

4.3.1. Model M5

$$(\Sigma_{k\in Mij} w_{ijku}) * u \leqslant \mathrm{Cmax} \quad \forall i, u, j = J_i \tag{5.1}$$

$$\Sigma_u w_{ijku} = p_{kij} * v_{ijk} \quad \forall i, j, \quad \forall k \in Mij \tag{5.2}$$

$$\Sigma_i \Sigma_j w_{ijku} \leqslant 1 \quad \forall k, u \tag{5.3}$$

$$p_{kij} * (w_{ijku} - w_{ijku+1}) + \Sigma_{r=u+2}^{T} w_{ijkr} \leqslant p_{kij}$$

$$\forall i, j, k, u = 1, \ldots, T-2 \tag{5.4}$$

$$\Sigma_{k\in Mij} v_{ijk} = 1 \quad \forall i, j \tag{5.5}$$

$$p_{kij} * w_{hgku} \leqslant \Sigma_{r=1}^{u-1} w_{ijkr} + M * (1 - z_{ijhgk})$$

$$\forall i \leqslant h, \quad \forall j, g, u = 2, \ldots, T \quad \forall k \in Mij \cap Mhg \tag{5.6}$$

$$p_{khg} * w_{ijku} \leqslant \Sigma_{r=1}^{u-1} w_{hgkr} + M * (z_{ijhgk})$$

$$\forall i \leqslant h, \quad \forall j, g, u = 2, \ldots, T \quad \forall k \in Mij \cap Mhg \tag{5.7}$$

$$\Sigma_{k\in Mij} p_{kij} * v_{ijk} * \Sigma_{k\in Mij} w_{ij+1ku} \leqslant \Sigma_{r=1}^{u-1} \Sigma_{k\in Mij} w_{ijkr} * v_{ijk}$$

$$\forall i, j = 1, \ldots, Ji-1, \quad \forall u = 2, \ldots, T \tag{5.8}$$

$$w_{ijku} \in \{0-1\} \quad \forall i, j, k, u \tag{5.9}$$

$$z_{ijhgk} \in \{0, 1\} \quad \forall i \leqslant h, \forall j, g, \forall k \in Mij \cap Mhg \tag{5.9}$$

$$v_{ijk} \in \{0, 1\} \quad \forall i, j, k \tag{5.10}$$

Constraint (5.1) determines the makespan. Constraint (5.2) determines number of period (processing time) of O_{ij} if assigned on machine k. Constraint (5.3) ensures that at each period of each machine can only one operation can be performed. Constraint (5.4) is guarantee that each operation will not be interrupted before it is finished. Constraint (5.5) ensures that an operation is performed on one and only one machine. Constraint (5.5 and 5.6) is guarantee that operation O_{ij} and operation O_{hg} cannot be done at the same time on any machine in the set $Mij \cap Mhg$. Constraint (5.8) presents the precedence relationship of two consecutive operations.

5. Computational comparison of the formulations

We compared these five mathematical models in terms of objective Cmax, CPU time, number of variables and constraints. For comparison randomly generated test problems by Fattahi et al. [14] are used. They divided test problems into two categories: small size FJSPs (SFJS1-SFJS10) and medium-large size FJSPs (MFJS1-MFJS10). They defined test problems by the size of the problem using i, j, k indices which means relatively number of jobs, operations and machines.

All five model was coded in the mathematical language GAMS and used CPLEX (for linear models) and SNOPT (for non-linear models) solvers. Test problems are run on PC with Core(TM) 2 Quard CPU, 2.66 GHz processor and 4 GB RAM. The runs are terminated after 3600 s. Comparisons of four performance measurements are presented in Tables 3–5 and Fig. 1–5

In terms of objective function Cmax, M2 is superior to the others. Table 2 represents that for small sized problems all models can find optimum solution except M5. Models cannot obtain optimum solution as the problems size increases in the limited time due to the increasing number of equations and variable as seen in Fig. 4 and 5. For Model M5, scheduling horizon was divided into time periods 1, 2, ..., T where T may be estimated by any technique and time interval is chosen by

Table 2
The size of test problems.

		i	j	k			i	j	k
Small size	SFJS1	2	2	2	Medium and large size	MFSJ1	5	3	6
	SFJS2	2	2	2		MFSJ2	5	3	7
	SFJS3	3	2	2		MFSJ3	6	3	7
	SFJS4	3	2	2		MFSJ4	7	3	7
	SFJS5	3	2	2		MFSJ5	7	3	7
	SFJS6	3	3	2		MFSJ6	8	3	7
	SFJS7	3	3	5		MFSJ7	8	4	7
	SFJS8	3	3	4		MFSJ8	9	4	8
	SFJS9	3	3	3		MFSJ9	11	4	8
	SFJS10	4	3	5		MFSJ10	12	4	8

Table 3
Comparison of Cmax.

		Sequence-position	Precedence			Time-indexed			Sequence-position	Precedence			Time-indexed
		M1	M2	M3	M4	M5			M1	M2	M3	M4	M5
Small size	SFJS1	66	66	66	66	80	Medium and large size	MFSJ1	530*	468	468*	468	x
	SFJS2	107	107	107	107	120		MFSJ2	496*	446	460*	446	x
	SFJS3	221	221	221	221	240		MFSJ3	x	466	x	466	x
	SFJS4	355	355	355	355	370		MFSJ4	x	564	x	590*	x
	SFJS5	119	119	119	119	140		MFSJ5	x	514	597*	546*	x
	SFJS6	320	320	320	320	x		MFSJ6	x	634	x	666*	x
	SFJS7	397	397	397	397	x		MFSJ7	x	928*	x	1990*	x
	SFJS8	253	253	253	253	x		MFSJ8	x	x	x	x	x
	SFJS9	210	210	210	210	210		MFSJ9	x	x	x	x	x
	SFJS10	526*	516	516	516	x		MFSJ10	x	x	x	x	x

x – Any feasible solution could not be found.
* Feasible but not optimum solution.

Table 4
CPU time comparison of test problems.

		Sequence-position	Precedence			Time-indexed			Sequence-position	Precedence			Time-indexed
		M1	M2	M3	M4	M5			M1	M2	M3	M4	M5
Small size	SFJS1	0.14	0.03	0.05	0.02	0.66	Medium and large size	MFSJ1	3600	0.78	3600	67	3600
	SFJS2	0.09	0.10	0.02	0.05	0.86		MFSJ2	3600	49	3600	117	3600
	SFJS3	10	0.05	0.48	0.25	269		MFSJ3	3600	191	3600	1685	3600
	SFJS4	127	0.04	1.12	0.27	676		MFSJ4	3600	1051	3600	3600	3600
	SFJS5	684	0.06	2.09	0.72	76		MFSJ5	3600	225	3600	3600	3600
	SFJS6	2652	0.28	13.30	1.11	3600		MFSJ6	3600	231	3600	3600	3600
	SFJS7	3600	0.03	1.23	0.39	3600		MFSJ7	3600	3600	3600	3600	3600
	SFJS8	3600	0.16	6.22	3.96	3600		MFSJ8	3600	3600	3600	3600	3600
	SFJS9	3600	1.26	47.00	3.21	3600		MFSJ9	3600	3600	3600	3600	3600
	SFJS10	3600	0.06	4.00	1.36	3600		MFSJ10	3600	3600	3600	3600	3600

researcher. In this paper we chose 10 time unit as interval for all test problems. It is assumed that succeeding operations can only start at the beginning of the time period even if the proceeding operation finishes earlier [36]. For instance as seen in Fig. 6, even operation of O_{11} ends in 47th time unit, succeeding operation O_{12} can start earliest at the beginning of fifth period (50th time unit). This is the reason of differences between optimal solutions of Model M5 and the others even test problems are solved on the same grounds.

Considering time intervals between periods in time-indexed scheduling approach has great impact on optimum solution and computational requirements. It means to obtain sensitive optimum solution; CPU time, number of equation and variable grow exponentially as the time interval increases as seen in the Fig. 1 and 2.

Table 5
Comparison of overall performance measurements.

Formulations		Performance measurements	SFJS1	SFJS2	SFJS3	SFJS4	SFJS5	SFJS6	SFJS7	SFJS8	SFJS9	SFJS10	MFSJ1	MFSJ2	MFSJ3	MFSJ4	MFSJ5	MFSJ6	MFSJ7	MFSJ8	MFSJ9	MFSJ10
Sequence-position	Model	Solution	66	107	221	355	119	320	397	253	210	526*	530*	496*	–	–	–	–	–	–	–	–
variable	M1	Found at node	5	3	272	316	16.490	11716	21	18710	4794	4200	970	1744	92	21	35	13	4	4	0	1
		Absolute gap	0	0	52.99	3.55	1.19	3.2	0	36.99	0	99	127	100	–	–	–	–	–	–	–	–
		Relative gap	0	0	0.315	0	0	0	0	0.17	0	0.23	0.32	0.25	–	–	–	–	–	–	–	–
		CPU	0.14	0.09	10	127	684	2652	3600	3600	559	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600
		Constraints	124	98	220	258	258	558	632	623	726	1576	3447	4014	5955	8274	8274	10460	13900	13900	43044	51568
		Variables	57	47	95	109	109	226	264	255	286	605	1273	1480	2158	2962	2962	3717	4909	4909	14841	17729
		Nonlinear entries	48	32	96	120	120	270	270	288	378	840	1980	2310	3528	4998	4998	6384	8512	8512	27456	33024
Precedence variable	Model	Solution	66	107	221	355	119	320	397	253	210	516	468	446	466	564	514	634	928*	–	–	–
	M2	Found at node	0	0	35	52	135	110	1	455	401	29	3891	4230	17781	2422503	871770	535047	5251180	13595	6725	3421
		Absolute gap	0	0	0	0	0	3.19	0	0	2.09	0	0	4.46	4.66	0	0	0	–	–	–	–
		Relative gap	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	–	–	–	–
		CPU	0.03	0.10	0.05	0.04	0.06	0.28	0.03	0.16	1.26	0.06	0.78	49	191	1051	225	231	3600	3600	3600	3600
		Constraints	42	34	71	71	87	123	147	147	149	204	355	423	578	731	719	882	1314	1523	2099	2450
		Variables	35	30	55	55	64	105	148	133	119	200	327	388	501	613	608	725	1038	1253	1650	1880
		Nonlinear entries	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Precedence variable	Model	Solution	66	107	221	355	119	320	397	253	210	516	468*	460*	–	–	597*	–	–	–	–	–
	M3	Found at node	pp	pp	1	72	61	130	pp	120	373	51	11532	11328	449	262	2206	177	38	53	33	6
		Absolute gap	6.6	0	2.21	3.54	1.19	3.20	3.97	2.52	2.09	5.16	–	174	–	–	274	–	–	–	–	–
		Relative gap	0	0	0	0	0	0	0	0	0	0	–	0	–	–	0.85	–	–	–	–	–
		CPU	0.1	0	0.48	1.12	2.09	13.32	1.23	6.22	47	3.59	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600
		Constraints	50	31	80	83	111	130	120	141	174	153	320	390	579	755	743	934	1511	1662	2384	2841
		Variables	37	24	58	60	79	97	94	106	126	120	240	290	422	545	537	670	1077	1194	1694	2008
		Nonlinear entries	96	48	160	168	240	256	192	264	368	240	608	768	1216	1632	1600	2056	3424	3680	5440	6576
	Model	Solution	66	107	221	355	119	320	397	253	210	516	468	446	466	590*	546*	666*	1990*	–	–	–
	M4	Found at node	pp	5	17	46	92	58	15	81	70	66	1118	1316	10050	7422	10250	6456	1236	605	186	186
		Absolute gap	0	0	2.21	3.5	1.19	3.20	0	2.53	0	0	0	4.46	4.66	94	99.65	52	1226	–	–	–
		Relative gap	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.22	0.084	1.60	–	–	–
		CPU	0	0.1	0.25	0.27	0.72	1.11	0.39	3.96	3.21	1.36	67	117	1685	3600	3600	3600	3600	3600	3600	3600
		Constraints	54	36	87	90	117	147	141	159	189	182	358	433	630	815	803	1003	1596	1767	2513	2513
		Variables	45	33	71	73	91	128	148	148	156	193	348	418	575	724	716	875	1337	1533	2109	2109
		Nonlinear entries	40	28	64	66	84	118	138	138	146	180	332	402	556	702	694	850	1304	1496	2064	2064
Time-indexed	Model	Solution	8	12	24	37	14	–	–	–	21	–	–	–	–	–	–	–	–	–	–	–
variable	M5	Found at node	1	5	73	196	22	–	–	–	597	–	–	–	–	–	–	–	–	–	–	–
		Absolute gap	0	0	2	4	0	–	–	–	0	–	–	–	–	–	–	–	–	–	–	–
		Relative gap	0	0	0	0	0	–	–	–	0	–	–	–	–	–	–	–	–	–	–	–
		CPU	0.66	0.86	269	676	76	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600
		Constraints	248	270	1257	2037	963	3396	4170	2985	2718	7828	13523	16431	23443	33161	32452	47474	72834	82513	117114	138762
		Variables	89	115	327	507	217	1120	2062	1126	731	3646	5054	5904	7121	9071	9064	12068	16156	20680	25392	27780
		Nonlinear entries	208	318	1764	4329	798	8892	13728	6264	4320	26137	37098	43794	53784	73455	69797	104604	165600	183816	219420	238464

Nonlinear entries: the number of nonlinear matrix entries in the model.
* Feasible but not optimum.

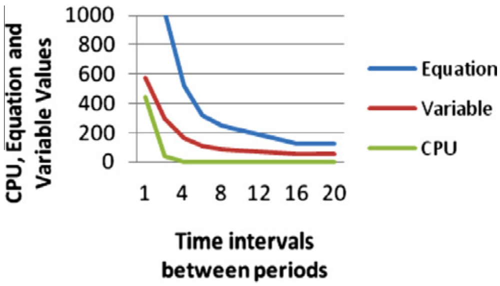

Fig. 1. Computational requirements for different time intervals.

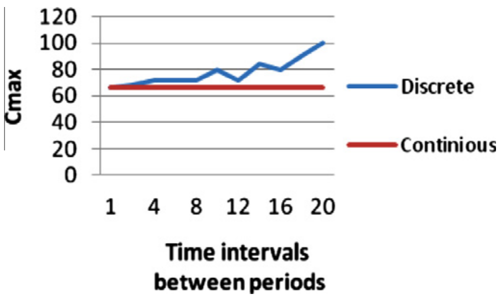

Fig. 2. Cmax exchange for discrete and continuous time model.

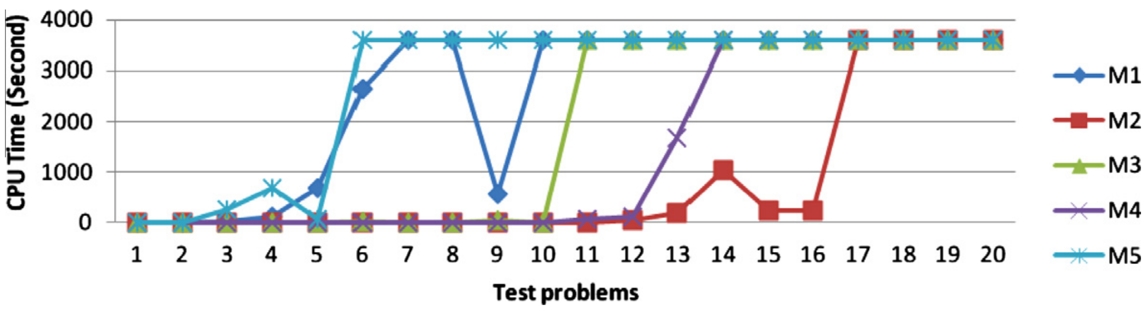

Fig. 3. Comparison of CPU time for test problems.

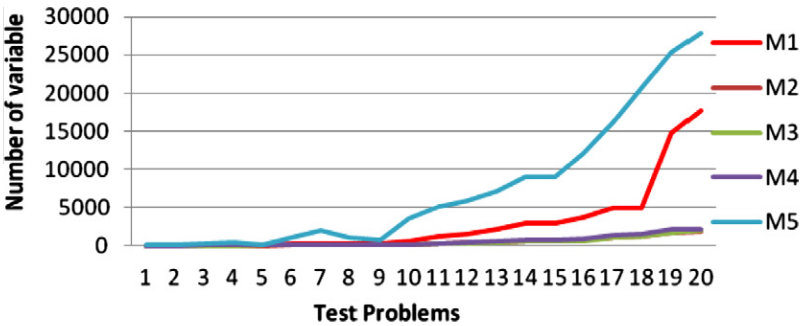

Fig. 4. Comparison of number of variables.

Let us handle SFJS1 test problem with optimum solution 66 time units. When each time period is considered 1 time unit for discrete time model, time-indexed model produces as the same value as the other continuous modeling approaches (Fig. 2). But this causes huge number of variable and equation and also high CPU time. In contrary, large time intervals are resulting faster but causes deviation from optimum solution obtained models that have continuous scheduling horizon.

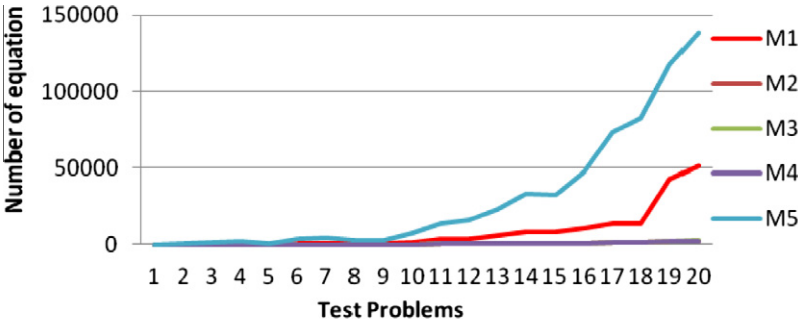

Fig. 5. Comparison of number of equations.

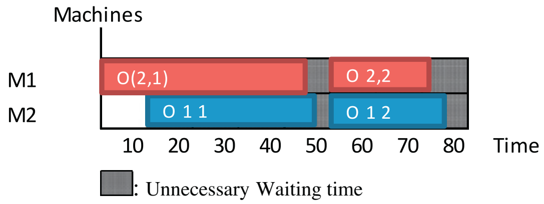

Fig. 6. Scheduling result of test problem SFJS1.

The second performance criterion is CPU times. As seen in the Table 3 and Fig. 3, Model M2 is the best in terms of CPU time. M4 is the second, M3 is third, M4 is fourth and M5 is last.

As shown in Fig. 4 and 5, number of variables and equations of models of M2, M3, M4 overlap. Even they have approximately the same number of variables and equations and they are in same modeling category, in terms of CPU time there occurs significant difference between M2 and the other two models especially in medium-large sized problems due to the linear structure of model M2.

6. Conclusion and suggestions

In this paper, literature is filtered and previously developed mathematical models are examined based on binary variables that they rely on for using of sequencing operations on machines. Review of the literature revealed that there exist three different binary variables developed by Wagner, Manne and Bowman are used for this aim. Based on these three different types of binary variables five different mathematical formulations are presented for FJSP with makespan performance measure.

Models are applied on different sized test problems. For test problems as the number of operations and machines increase, computation time increases exponentially. Additionally for time-indexed based model to obtain optimum solution time intervals should be considered as small as possible even each time unit must equal to a second. This case increases computation time enormously and causes to be the worst computation timed model. The least computation time is obtained by Manne's precedence variable based formulation and among this approach M2 is the model with least computation time for almost all optimally solved test problems. Based on the results obtained in this paper, it is recommended to use precedence variable based models among them especially model M2 for FJSP.

Also it has been showed in this paper there formulated various versions of FJSP by researchers such as with set up time, buffer size constraint, multi-objective etc. But it has not been paid enough attention FJSP with lot streaming consideration. Future research may be conducted to develop mathematical models for FJSP considering lot streaming with all classes and with different versions such as with set up time, buffer size, transportation time.

References

[1] M. Pinedo, Scheduling: Theory, Algorithms and Systems, Prentice-Hall, New Jersey, Englewood Cliffs, 1995.
[2] M.R. Garey, D.S. Johnson, R. Sethi, The complexity of flowshop and jobshop scheduling scheduling, Math. Oper. Res. 1 (2) (1976) 117–129.
[3] A. Bagheri, M. Zandieh, I. Mahdavi, M. Yazdani, An artificial immune algorithm for the flexible job-shop scheduling problem, Future Generation Comput. Syst. 26 (4) (2010) 533–541.
[4] C. Özgüven, L. Özbakır, Y. Yavuz, Mathematical models for job-shop scheduling problems with routing and process plan flexibility, Appl. Math. Modell. 34 (2010) 1539–1548.
[5] H.M. Wagner, An integer liinear programming model for machine scheduling, Naval Res. Logistics Quart. 6 (1959) 131–140.
[6] E.H. Bowman, The scheduling sequence problem, Oper. Res. 7 (1959) 621–624.
[7] A.S. Manne, On the job-shop scheduling problem, Oper. Res. 8 (1960) 219–223.

[8] K.R. Baker, D. Trietsch, Principles of Sequencing and Scheduling, Wiley, New Jersey, 2009.
[9] Y. Unlu, S.J. Mason, Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems, Comput. Ind. Eng. 58 (2010) 785–800.
[10] C.-H. Pan, A study of integer programming formulations for scheduling problems, Int. J. Syst. Sci. 28 (1997) 33–41.
[11] J. Blazewicz, M. Dror, J. Weglarz, Mathematical programming formulations for machine scheduling: a survey, Eur. J. Oper. Res. 51 (1991) 283–300.
[12] A.B. Keha, K. Ketan, W. Fowler, Mixed integer programming formulations for single machine scheduling problems, Comput. Ind. Eng. 56 (2009) 357–367.
[13] J. Pan, J. Chen, Mixed binary integer programming formulations for the reentrant job shop scheduling problem, Comput. Oper. Res. 32 (2005) 1197–1212.
[14] P. Fattahi, M.S. Mehrebad, F. Jolai, Mathematical modeling and heuristic approaches to flexible job shop scheduling problems, J. Intell. Manuf. 18 (2007) 331–342.
[15] G. Zhang, L. Gao, Y. Shi, An effective genetic algorithm for the flexible job-shop scheduling problem, Expert Syst. Appl. 38 (4) (2011) 3563–3573.
[16] Q. Zhang, H. Manier, M.-A. Manier, A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times, Comput. Oper. Res. 39 (2012) 1713–1723.
[17] G. Zhang, X. Shao, P. Li, L. Gao, An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, Comput. Ind. Eng. 56 (2009) 1309–1318.
[18] E. Moradi, S.M.T. Fatemi Ghomi, M. Zandieh, Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem, Expert Syst. Appl. 38 (2011) 7169–7178.
[19] P. Fattahi, A. Fallahi, Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability, CIRP J. Manuf. Sci. Technol. 2 (2010) 114–123.
[20] Y.H. Lee, C.S. Jeong, C. Moon, Advanced planning and scheduling with outsourcing in manufacturing supply chain, Comput. Ind. Eng. 43 (2002) 351–374.
[21] M.S. Mehrabad, P. Fattahi, Flexible job shop scheduling with tabu search algorithms, Int. J. Adv. Manuf. Technol. 32 (2007) 563–570.
[22] C.S. Thomalla, Job shop scheduling with alternative process plans, Int. J. Production Economics 71 (2001) 125–134.
[23] C.Y. Low, T.H. Wu, Mathematical modelling and heuristic approaches to operation scheduling problems in an FMS environment, Int. J. Prod. Res. 39 (2001) 689–708.
[24] N. Imanipour, Modeling & solving flexible job shop problem with sequence dependent setup times, in: Proceedings of the International conference on service systems and service management, October 25–27, 2006, vol. 2, IEEE, 2006, pp. 1205–1210.
[25] H. Tamaki, T. Ono, H. Murao, S. Kitamura, Modeling and genetic solution of a class of flexible job shop scheduling problems, in: Proceedings of the IEEE Symposium on Emerging Techonologies and Factory Automation, vol. 2, IEEE, 2001, pp. 343–350.
[26] C. Low, Y. Yip, T.H. Wu, Modeling and heuristics of FMS scheduling with multiple objectives, Comput. Oper. Res. 33 (2006) 674–694.
[27] L. Lin, H. Jia-zhen, Multi-objective flexible job-shop scheduling problem in steel tubes production, SETP 29 (8) (2009) 117–126.
[28] M. Gen, J. Gao, L. Lin, Multistage-Based genetic algorithm for flexible job-shop scheduling problem, Intell. Evolut. Syst. 187 (2009) 183–196.
[29] M.C. Gomes, A.P. Barbosa PÓ VOA, A.Q. Novais, Optimal scheduling for flexible job shop operation, Int. J. Prod. Res. 43 (2005) 2323–2353.
[30] K.-H. Kim, P.J. Egbelu, Scheduling in a production environment with multiple process plans per job, Int. J. Prod. Res. 37 (1999) 2725–2753.
[31] J. Gao, M. Gen, L. Sun, Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm, J. Intell. Manuf. 17 (2006) 493–507.
[32] S.A. Torabi, B. Karimi, S.M.T. Fatemi Ghomi, The common cycle economic lot scheduling in flexible job shops: the finite horizon case, Int. J. Production Economics 97 (2005) 52–65.
[33] P. Fattahi, F. Jolai, J. Arkat, Flexible job shop scheduling with overlapping in operations, Appl. Math. Modell. 33 (2009) 3076–3087.
[34] M.A. Khalife, B. Abbasi, A.H.K.D. Abadi, A simulated annealing algorithm for multi objective flexible job shop scheduling with overlapping in operations, J. Ind. Eng. 5 (2010) 17–28.
[35] J. Liu, B.L. MacCarty, A global milp model for FMS scheduling, Eur. J. Oper. Res. 100 (1997) 441–453.
[36] S.E. Kesen, Z. Güngör, How important is the batch splitting activity in scheduling of virtual manufacturing cells (VMCs)?, Int J. Prod. Res. 49 (2011) 1645–1667.
[37] S.E. Kesen, S.K. Das, Z. Güngör, A genetic algorithm based heuristic for scheduling of virtual manufacturing cells (VMCs), Comput. Oper. Res. 37 (2010) 1148–1156.