Journal of Industrial and Systems Engineering
Vol. 11, No.2, pp. 134-150
Spring (April) 2018

Minimizing the maximum tardiness and makespan criteria in a job shop scheduling problem with sequence dependent setup times

Mahdi Heydari[1*], Adel Aazami[1]

[1]*School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran*

mheydari@iust.ac.ir, a_aazami@ind.iust.ac.ir

Abstract

The job shop scheduling problem (JSP) is one of the most difficult problems in traditional scheduling because any job consists of a set operations and also any operation processes by a machine. Whereas the operation is placed in the machine, it is essential to be considering setup times that the times strongly depend on the various sequencing of jobs on the machines. This research is developed a two-objective model to solve JSP with sequence-dependent setup times (SDST). Considering SDST and optimizing of the both objectives simultaneously (makespan and maximum tardiness) bring us closer to natural-world problems. The ε-constraint method is applied to solve the mentioned tow-objective model. A set of numerical data is generated and tested to validate the model's efficiency and flexibility. The developed model can efficiently use for solving JSPs in the real world, especially for manufacturing companies with having setup and delivery time's constraints.

Keywords: Job shop scheduling, sequence-dependent setup times, makespan criterion, maximum tardiness criterion, mixed integer nonlinear programming

1- Introduction

Scheduling refers to the process of allocating operations to machines at certain intervals. A scheduling problem can be classified based on four parameters: the arrival patterns of jobs, number of machines in the shop, flow patterns, and scheduling criterion according to which the scheduling is evaluated. From a variety of scheduling problems, job shop scheduling problem (JSP) is of great importance. A JSP is defined as "a set of *n* jobs available to be processed by *m* machine, each one with specified technical limits". The aim is to find a job sequence on the machine as to meet technical limits and optimize the sequence according to some performance criteria (Tan and Khoshnevis 2000). JSPs are one of the most difficult combinatorial optimization problems, the NP-complete problems, as suggested by the research Garey et al. (1976).

In many manufacturing companies such as automobile production systems, pharmaceutical companies, printing industries, and chemical manufacturers; setup times for operations like cleaning and replacing the equipment are heavily dependent on the sequencing of operations on machines. As a result, taking sequence-dependent setup times (SDST) into account may assist to obtain a more accurate scheduling plan. In real scheduling problems, however, a significant savings obtain in job completion and tardiness times if an optimal sequence is achieved by considering SDST.

*Corresponding author
ISSN: 1735-8272,

In this research, the first and most important innovation element includes the considering of sequence-dependent setup times for modeling the scheduling problem, in addition to job completion times.

The second element is the application of a two-objective approach for problem modeling and solution, so both criteria – makespan and maximum tardiness – are simultaneously minimized. The research aims to propose a model for JSPs with considering SDST, in order to simultaneously minimize two criteria. The model shows a high efficiency to solve real-world JSPs; especially for manufacturing companies with having setup times and delivery lead time constraints.

The study is organized as the following. Section 2 provides an overview of the literature on the job shop scheduling and flexible job shop scheduling problems (FJSPs). Section 3 presents the mathematical model for a two-objective JSP with considering sequence-dependent setup times. Section 4 evaluates the model flexibility and performance based on a set of numerical data. Finally, the conclusion and some avenue for future researches are presented.

2- Literature review

Many studies have been widely investigated JSPs and suggested a variety of optimization and approximation algorithms. Optimization algorithms which are principally based on the branch and bound approach (e.g. Carlier and Pinson (1989); Brucker et al. (1994)) demonstrate successful performance for small-sized problems, while they fail to solve problems larger than 250 operations in a reasonable time. However, approximation algorithms provide more accurate answers for JSP, such as priority dispatch, shifting bottleneck approach, meta-heuristic methods. A number of meta-heuristic methods like genetic algorithm (GA) (Della Croce et al. 1995), simulated annealing (SA) (Laarhoven et al. 1992), tabu search (TS) (Taillard (1994); Nowicki and Smutnicki (1996); Zhang et al. (2007) and Zhang et al. (2008)) can obtain high-quality solutions in a more reasonable time, and hence they have been taken into account by many researchers.

Kim and Bobrowski (1994) investigated the efficacy of sequence-dependent setup times on the JSP performance. They arranged and examined scheduling rules by considering whether setup time or due date data is employed. Lee and Pinedo (1997) studied on the parallel machines scheduling with sequence-dependent setup times. They assumed a number of jobs to be performed on a number of identical machines in parallel and also offered a three phase heuristic for minimizing the sum of the weighted tardiness. Pinedo and Singer (1999) presented a shifting bottleneck heuristic for the minimizing total weighted tardiness in the JSP. They proposed the method that decomposed the JSP into a number of single machine sub problems. Sun and Noble (1999) investigated the JSP with release dates, due dates, and SDST with the scheduling objective to minimize the weighted sum of squared tardiness. Ponnambalam et al. (2000) developed a Tabu search algorithm to solve classic JSP where each operation was performed by a predetermined machine. Based on a tree search procedure, Asano and Ohta (2002) provided a heuristic method for the JSP to minimize total weighted tardiness. In the research, JSP to minimize the maximum tardiness was limited to particular sub-set of schedules determined at each vertex along the search tree. Park et al. (2003) described a hybrid genetic algorithm for JSP. In order to provide the solution, the authors used the same method developed by Ponnambalam et al. (2000). Some researchers solved JSPs using genetic algorithms (Qi et al. 2000, Watanabe et al. 2005). Mattfeld and Bierwirth (2004) suggested a genetic algorithm for JSP with tardiness objectives. Release times and due dates were considered for operations and jobs, respectively, and the total weighted tardiness was the objective function. For these problems, Gonçalves et al. (2005) studied a hybrid GA with the aim of minimizing the makespan criterion.

Fattahi et al. (2007) and Saidi-Mehrabad and Fattahi (2007) investigated a flexible job shop scheduling problem (FJSP). Given it was an NP-hard problem; the authors developed a Tabu search algorithm based on a hierarchical approach. Petrovic et al. (2008) discussed on a fuzzy JSP with lot-sizing and developed a fuzzy rule-based system which determined the size of lots by the following premise variables: the static slack of the job, size of the job, and etc. Naderi et al. (2009) considered JSP where the setup times were sequence dependent under minimization of the maximum makespan. They proposed a hybridized genetic algorithm with a diversification mechanism to solve the complicated problem. They also reviewed different parameters of the genetic algorithm to calibrate the algorithm by using the Taguchi method. Naderi et al. (2009) evaluated scheduling dependent setup time job shops with preventive maintenance and machine availability constraints. The authors used both simulated annealing and genetic algorithms to minimize the makespan criterion. Yang et al. (2010) offered an improved constraint satisfaction adaptive neural network for the JSP that integrated several heuristics within the neural network.

To solve FJSP, Al-Hinai and ElMekkawy (2011a) suggested an algorithm for machine failure events where the levels of failure were not studied. In other study, the authors (Al-Hinai and ElMekkawy 2011b) investigated the robust FJSP with possible failure times, by using a hybrid genetic algorithm. Bagheri and Zandieh (2011) studied bi-critria FJSP with SDST to minimize makespan and mean tardiness. They proposed a variable neighborhood search (VNS) algorithm based on integrated approach to solve the problem. Pinedo (2012) investigated the JSP and its models in the seventh chapter of his book. He formulated the disjunctive programming for the JSP and applied the branch and bound procedure to problem. Özgüven et al. (2012) developed mixed integer goal programming models for the FJSP with separable and non-separable SDST. Zhang et al. (2012) introduced a mathematical model for FJSP to minimize makespan criterion with transportation constraints and bounded processing times. Ebadi and Moslehi (2013) proposed an optimum method for preemptive/preventative JSP, where a robust solution methodology was developed to minimize makespan criterion. Mousakhani (2013) studied the FJSP with setup times where the setups were sequence-dependent. He proposed an effective meta-heuristic algorithm based on iterated local search and compared with a Tabu search. González et al. (2013) investigated the JSP with SDST and maximum lateness minimization by means of a Tabu search algorithm and defined the new local search neighborhood structure.

Li et al. (2014) presented a discrete artificial bee colony algorithm for the multi-objective FJSP with maintenance activities. They considered the maximum completion time– namely, makespan, the total workload of machines and the workload of the critical machine as the performance criteria. Naderi and Azab (2014) evaluated modeling and solutions for scheduling of distributed job shops and developed the classic single-facility JSP to multi-facility problem. Tan et al. (2015) answered to two fundamental questions concerning the usage of the shifting bottleneck (SB) procedure (Configuration and the advantages of the SB) for optimizing the total weighted tardiness criterion for the classical JSP. Sharma and Jain (2015) analyzed the dispatching rules in a stochastic dynamic job shop manufacturing system with SDST, using the simulation procedures. The authors revealed that stochastic dynamic JSPs with SDST are from the difficult scheduling problems.

For the JSPs, Kurdi (2016) developed an effective new model where an evolutionary model and a new selective migration mechanism inspired by the nature were suggested as to improve search diversification and to delay premature convergence. Kuhpfahl and Bierwirth (2016) studied the local search neighborhoods for the JSP to minimize the total weighted tardiness. Based on disjunctive graphs, the authors developed an approach in order to capture the overall structure of neighborhoods. Fattahi and Daneshamooz (2017) considered a JSP with a parallel assembly stage and lot streaming for the first time in machining and assembly stages. They proposed four hybrid algorithms based on iterative procedures for solving the problem in medium and large dimensions. As the most recent, but certainly not the last developed research, Shen et al. (2017) addressed the FJSP with SDST so that the objective was to minimize makespan. They first introduced a mathematical model which could solve small examples to optimality, and then developed a Tabu search algorithm with a diversification structure and specific neighborhood functions. A summary of the most important reviewed studies is presented in table 1.

Table 1. Review of recent relevant studies along with the present study

Researchers	Presenting a mathematical model	SDST	Objective function				Solution approach		
			makespan	Maximum tardiness	Total tardiness	Total workload	Exact	Heuristic	Meta-heuristic
Bagheri and Zandieh (2011)		*	*		*				VNS
Pinedo (2012)			*		*		Branch and bound	Shifting bottleneck heuristic	
Zhang et al. (2012)	*		*						GA and TS
González et al. (2013)		*			*				TS
Li et al. (2014)	*		*			*			TS
Sharma and Jain (2015)		*	*	*				Simulation approach	
Kuhpfahl and Bierwirth (2016)					*			Neighborhood search	
Fattahi and Daneshamooz (2017)	*		*						GA and SA
Shen et al. (2017)	* (Not Perfect)	*	*						TS
This paper	*	*	*	*			ε-Constraint		

According to the literature review on JSPs and Table 1, there is a very little research investigated the problems with considering sequence-dependent setup times. Furthermore, no model has been yet developed with considering SDST, in particular to reduce makespan criterion and minimize the maximum tardiness simultaneously. We will close to the real world conditions with a higher efficiency if the scheduling problems are investigated by using the two objectives. Therefore, the body of literature is very thin in this part and the current study aims to cover this research gap.

3- Mathematical modeling of the problem

3-1- Problem description

The proposed model aims to find the optimal sequence of different operations on machines in a two-objective JSP with considering sequence-dependent setup times. In this model, the first objective function is to minimize the makespan criterion, which has been widely studied in the literature. The second objective function has to minimize the maximum tardiness time; however, appear to have been less investigated. Considering SDST and optimizing both objectives simultaneously make us closer to real-world scheduling problems. The presented model and results can be used to solve JSPs for manufacturing companies with different stages and operations. By solving the model, finally, the production management can determine the completion time, tardiness time, and two above-mentioned important criteria (the objectives).

The following assumptions are considered to model the two-objective JSP with the SDST:

- All jobs are available at the beginning of manufacturing scheduling.
- Operation cut is not permitted.
- Each machine can only perform a single operation of a job at a certain time.
- Each operation is assigned to one priority of a certain machine.

3-2- Notations

Indexes:

I	Machines
J, J'	Jobs
K, K'	Operations
L	Priority

Parameters:

k_j	The number of operations for job j
$k'_{j'}$	The number of operations for job *j'*
l_i	The number of priorities/jobs to be placed on the machine *i*
n	The total number of jobs
m	The total number of machines
P_{jk}	Duration of the k^{th} operation of the j^{th} job
d_j	Delivery time of the j^{th} job
$S_{ijkj'k'}$	Setup time of the i^{th} machine if the operation *k'* of the job *j'* follows the operation *k* of the job *j*
$S_{i00j'k'}$	Setup time of the i^{th} machine if the operation *k'* of the job *j'* is placed on the first priority; i.e., after the 0^{th} dummy operation of the 0^{th} dummy job
M	Positive large value
x_{ijk}	$\begin{cases} 1 & \text{If the } k^{th} \text{ operation of the } j^{th} \text{ job is operated on the } i^{th} \text{ machine} \\ 0 & \text{Otherwise} \end{cases}$

Decision variables:

C_{max}	The completion time of the last job
T_{max}	The maximum tardiness
T_j	The tardiness time for the j^{th} job
TF_{il}	The completion time for the l^{th} priority on the i^{th} machine
C_{jk}	The completion time for the k^{th} operation of the j^{th} job
z_{ijkl}	$\begin{cases} 1 & \text{If the } k^{th} \text{ operation of the } j^{th} \text{ job is placed on the } l^{th} \text{ priority in the } i^{th} \text{ machine} \\ 0 & \text{Otherwise} \end{cases}$

3-3- Objective functions and constraints

The number of priorities to be operated by the machine *i* (parameter l_i) is obtained by using equation (1). Indeed, this equation presents the sum of all operations of all jobs that should be performed on the machine *i*.

$$l_i = \sum_{j=0}^{n} \sum_{k=0}^{k_j} x_{ijk} \quad for\ \ i = 1,2,\dots,m \tag{1}$$

The mathematical model for the two-objective JSP is developed as follows.

$$\text{Min}\ C_{max} \tag{2}$$

$$\text{Min}\ T_{max} \tag{3}$$

$Subject\ to$:

$$C_{max} \geq TF_{i,l_i} \quad for\ \ i = 1,2,\dots,m \tag{4}$$

$$T_{max} \geq T_{j'} \quad for\ \ j' = 1,2,\dots,n \tag{5}$$

$$T_{j'} \geq 0 \quad for\ \ j' = 1,2,\dots,n \tag{6}$$

$$T_{j'} \geq C_{j',k'_{j'}} - d_{j'} \quad for\ \ j' = 1,2,\dots,n \tag{7}$$

$$C_{j'k'} = \max\left\{C_{j',(k'-1)}\,, \sum_{i=1}^{m} \sum_{l=1}^{l_i} z_{ij'k'l}.TF_{i,(l-1)}\right\} + P_{j'k'} + \sum_{i=1}^{m} \sum_{j=0}^{n} \sum_{k=0}^{k_j} \sum_{l=1}^{l_i} z_{ijk,(l-1)}.z_{ij'k'l}.S_{ijkj'k'} \quad for\ \ j' = 1,2,\dots,n\ \ and\ \ k' = 1,2,\dots,k'_{j'} \tag{8}$$

$$\sum_{j=0}^{n}\sum_{k=0}^{k_j} z_{ijkl} = 1 \quad for\ i = 1,2,\dots,m\ and\ l = 1,2,\dots,l_i \tag{9}$$

$$\sum_{i=1}^{m}\sum_{l=1}^{l_i} x_{ijk}.z_{ijkl} = 1 \quad for\ j = 0,1,\dots,n\ and\ k = 0,1,\dots,k_j \tag{10}$$

$$\sum_{l=1}^{l_i} z_{ijkl} = x_{ijk} \quad for\ i = 1,2,\dots,m\ and\ j = 0,1,\dots,n\ and\ k = 0,1,\dots,k_j \tag{11}$$

$$TF_{il} = \max\left\{TF_{i,(l-1)}, \sum_{j'=1}^{n}\sum_{k'=1}^{k'_{j'}} x_{ij'k'}.z_{ij'k'l}.C_{j',(k'-1)}\right\} + \sum_{j'=1}^{n}\sum_{k'=1}^{k'_{j'}} P_{j'k'}.x_{ij'k'}.z_{ij'k'l} + \sum_{j=0}^{n}\sum_{j'=1}^{n}\sum_{k=0}^{k_j}\sum_{k'=1}^{k'_{j'}} z_{ijk,(l-1)}.z_{ij'k'l}.S_{ijkj'k'} \quad for\ i = 1,2,\dots,m\ and\ l = 1,2,\dots,l_i \tag{12}$$

$$TF_{i0} = 0 \quad for\ i = 1,2,\dots,m \tag{13}$$

$$z_{i000} = 1 \quad for\ i = 1,2,\dots,m \tag{14}$$

$$C_{00} = 0 \tag{15}$$

$$TF_{il}, C_{jk}, T_j \geq 0 \quad for\ i = 1,2,\dots,m\ and\ j = 0,1,\dots,n\ and\ k = 0,1,\dots,k_j\ and\ l = 1,2,\dots,l_i \tag{16}$$

$$z_{ijkl} = \{0,1\} \quad for\ i = 1,2,\dots,m\ and\ j = 0,1,\dots,n\ and\ k = 0,1,\dots,k_j\ and\ l = 1,2,\dots,l_i \tag{17}$$

Constraints (2) and (3) show the objective functions of the problem that are minimizing the makespan criterion and minimizing the maximum tardiness. Constraint (4) shows that the makespan criterion must be greater or equal the completion times of each jobs placed on the last priority in the i^{th} machine. Constraint (5) represents that the T_{max} criterion must be greater or equal the tardiness times for each job. However, constraint (6) ensures that tardiness of each job has a nonnegative value. Constraint (7) indicates that the tardiness of the j^{th} job must be greater or equal the completion time of the last operations of the j^{th} job, minus the delivery time of the j^{th} job.

Constraint (8) states that the amount of completion time for the operation *k*' of the job *j*' is obtained by the sum of three parts. The first is the maximum amount of between the completion time for the operation *k*'–*1* of the job *j*', and the completion time for the priority *l-1* on the machine *i*, provided that the operation *k*' of the job *j*' is placed on the priority *l*. The second part includes the duration of the operation *k*' of the j'^{th} job. Finally, the third part considers the setup time for the k'^{th} operation of the j'^{th} job, provided that the k'^{th} operation of the j'^{th} job on the i^{th} machine is placed on the next priority of the k^{th} operation of the j^{th} job, while the setup time of the machine is assumed to be performed on work piece after its arrival.

Constraint (9) indicates that only one single operation *k* of the job *j* can be placed on the l^{th} priority of the i^{th} machines, whereas constraint (10) shows that any k^{th} operation of the j^{th} job which can be processed on the i^{th} machine needs to be placed on a single priority *l*. Constraint (11) shows that only those jobs could be placed on the l^{th} priority that their k^{th} operation on the i^{th} machine is possible for performing.

Constraint (12) shows that the amount of completion time for the l^{th} priority on the i^{th} machine may be obtained from the sum of three parts. The first includes the maximum amount of between the completion time for the $l\text{-}1^{th}$ priority on the i^{th} machine, and the completion time for the $k'\text{–}1^{th}$ operation of the j'^{th} job, provided that the k'^{th} operation of the j'^{th} job is placed on the l^{th} priority. The second part is the duration of the k'^{th} operation of the j'^{th} job, provided that this operation is placed into the l^{th} priority in the i^{th} machine. Finally, the third part deals with the setup time for the k'^{th} operation of the j'^{th} job, provided that the k'^{th} operation of the j'^{th} job on the i^{th} machine is placed into the next priority of the k^{th} operation of the j^{th} job, while the setup times of the machine must be performed on work piece after its arrival.

Constraint (13) establishes a dummy priority with the operation duration of 0, in order to create a starting point for the model solution. This constraint shows that the completion times of zero priorities will be zero for all machines. Constraint (14) necessitates that the 0^{th} dummy operation of the 0^{th}

dummy job on the 0th dummy priority is placed on the i^{th} machine in order to start the process. Constraint (15) states that the completion time of such operation must be zero, because the duration for the 0th dummy operation of the 0th dummy job is equal to zero. Constraints (16) and (17) represent the decision variables of the problem.

In the proposed model, the setup times of the machine must be operated on work piece after its arrival. If the manufacturing process is performed in a fashion that the setup times of the machine can be operated before the arrival of work piece, then constraints (18) and (19) will replace constraints (8) and (12), respectively.

$$C_{j'k'} = \max\left\{C_{j',(k'-1)}, \sum_{i=1}^{m}\sum_{l=1}^{l_i} z_{ij'k'l}.TF_{i,(l-1)} + \sum_{i=1}^{m}\sum_{j=0}^{n}\sum_{k=0}^{k_j}\sum_{l=1}^{l_i} z_{ijk,(l-1)}.z_{ij'k'l}.S_{ijkj'k'}\right\} + P_{j'k'} \quad for\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'} \tag{18}$$

$$TF_{il} = \max\left\{TF_{i,(l-1)} + \sum_{j=0}^{n}\sum_{j'=1}^{n}\sum_{k=0}^{k_j}\sum_{k'=1}^{k'_{j'}} z_{ijk,(l-1)}.z_{ij'k'l}.S_{ijkj'k'}, \sum_{j'=1}^{n}\sum_{k'=1}^{k'_{j'}} x_{ij'k'}.z_{ij'k'l}.C_{j',(k'-1)}\right\} + \sum_{j'=1}^{n}\sum_{k'=1}^{k'_{j'}} P_{j'k'}.x_{ij'k'}.z_{ij'k'l} \quad for\ i = 1,2,\dots,m\ and\ l = 1,2,\dots,l_i \tag{19}$$

Constraint (18) shows that the amount of completion time for the k'^{th} operation of the j'^{th} job is composed of two parts. The first is the maximum amount of between the completion time for the k'–1^{th} operation of the j'^{th} job and the sum of completion times for the l-1^{th} priority on the i^{th} machine, provided that the k'^{th} operation of the j'^{th} job is placed into the l^{th} priority and the setup times for the i^{th} machine before the arrival of work piece; while the k'^{th} operation of the j'^{th} job is placed in next priority for the k^{th} operation of the j^{th} job. The second part includes the duration of the k'^{th} operation of the j'^{th} job.

Constraint (19) indicates that the amount of completion time for the l^{th} priority on the i^{th} machine may be obtained from the sum of two parts. The first is the maximum amount of between the completion time for the k'–1^{th} operation of the j'^{th} job, provided that the k'^{th} operation of the j'^{th} job is placed into the l^{th} priority on the i^{th} machine, and that the sum of completion times for the l-1^{th} priority on the i^{th} machine and the setup time for the i^{th} machine before the arrival of work piece; while the k'^{th} operation of the j'^{th} job is placed in next priority for the k^{th} operation of the j^{th} job. The second part includes the duration of the k'^{th} operation of the j'^{th} job, provided that this operation is placed into the l^{th} priority.

3-4- Linearization of non-linear programming model

As found, constraints (8) and (12) are nonlinear equation, due to the multiplication of two variables for each constraint. In order to solve the problem, these constraints must be converted into linear equation form.

Constraint (8) is divided into three parts as follows:

$$C_{j'k'} = f_{j'k'} + P_{j'k'} + v_{j'k'} \quad for\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'} \tag{20}$$

$$f_{j'k'} = \max\left\{C_{j',(k'-1)}, \sum_{i=1}^{m}\sum_{l=1}^{l_i} z_{ij'k'l}.TF_{i,(l-1)}\right\} \quad for\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'} \tag{21}$$

$$v_{j'k'} = \sum_{i=1}^{m}\sum_{j=0}^{n}\sum_{k=0}^{k_j}\sum_{l=1}^{l_i} z_{ijk,(l-1)}.z_{ij'k'l}.S_{ijkj'k'} \quad for\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'} \tag{22}$$

Constraint (21) can be written as constraints (23) and (24).

$$f_{j'k'} \geq C_{j',(k'-1)} \quad for\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'} \tag{23}$$

$$f_{j'k'} \geq \sum_{i=1}^{m} \sum_{l=1}^{l_i} z_{ij'k'l}.TF_{i,(l-1)} \quad for\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'} \tag{24}$$

Here, equation (24) shows a non-linear equation, where a binary variable is multiplied by a continuous variable. If assumed $O_{ij'k'l} = z_{ij'k'l}.TF_{i,(l-1)}$ then the linear form of this constraint is represented as Constraints (25) and (26).

$$O_{ij'k'l} \geq \left(z_{ij'k'l} - 1\right).M + TF_{i,(l-1)} \quad for\ i = 1,2,\dots,m\ and\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'}\ and\ l = 1,2,\dots,l_i \tag{25}$$

$$O_{ij'k'l} \geq z_{ij'k'l}.(-M) \quad for\ i = 1,2,\dots,m\ and\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'}\ and\ l = 1,2,\dots,l_i \tag{26}$$

Equation (22) is also a non-linear equation where two binary variables are multiplied. If assumed $Q_{ijkj'k'l} = z_{ijk,(l-1)}.z_{ij'k'l}.S_{ijkj'k'}$ then this non-linear equation is converted into a linear equation form by constraints (27) and (28).

$$Q_{ijkj'k'l} \geq \frac{1}{2} \times S_{ijkj'k'} \times \left(z_{ijk,(l-1)} + z_{ij'k'l}\right) + M \times \left(z_{ijk,(l-1)} + z_{ij'k'l} - 2\right) \quad for\ i = 1,2,\dots,m\ and\ j = 0,1,\dots,n\ and\ k = 0,1,\dots,k_j\ and\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'}\ and\ l = 1,2,\dots,l_i \tag{27}$$

$$Q_{ijkj'k'l} \geq 0 \quad for\ i = 1,2,\dots,m\ and\ j = 0,1,\dots,n\ and\ k = 0,1,\dots,k_j\ and\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'}\ and\ l = 1,2,\dots,l_i \tag{28}$$

Now, constraint (12) is divided into three parts as follows:

$$TF_{il} = h_{il} + \sum_{j'=1}^{n} \sum_{k'=1}^{k'_{j'}} P_{j'k'}.x_{ij'k'}.z_{ij'k'l} + w_{il} \quad for\ i = 1,2,\dots,m\ and\ l = 1,2,\dots,l_i \tag{29}$$

$$h_{il} = \max\left\{TF_{i,(l-1)}, \sum_{j'=1}^{n} \sum_{k'=1}^{k'_{j'}} x_{ij'k'}.z_{ij'k'l}.C_{j',(k'-1)}\right\} \quad for\ i = 1,2,\dots,m\ and\ l = 1,2,\dots,l_i \tag{30}$$

$$w_{il} = \sum_{j=0}^{n} \sum_{j'=1}^{n} \sum_{k=0}^{k_j} \sum_{k'=1}^{k'_{j'}} z_{ijk,(l-1)}.z_{ij'k'l}.S_{ijkj'k'} \quad for\ i = 1,2,\dots,m\ and\ l = 1,2,\dots,l_i \tag{31}$$

Constraint (30) can be written as constraints (32) and (33).

$$h_{il} \geq TF_{i,(l-1)} \quad for\ i = 1,2,\dots,m\ and\ l = 1,2,\dots,l_i \tag{32}$$

$$h_{il} \geq \sum_{j'=1}^{n} \sum_{k'=1}^{k'_{j'}} x_{ij'k'}.z_{ij'k'l}.C_{j',(k'-1)} \quad for\ i = 1,2,\dots,m\ and\ l = 1,2,\dots,l_i \tag{33}$$

Here, equation (33) is a non-linear equation, where a binary variable is multiplied by a continuous variable. If assumed $R_{ij'k'l} = x_{ij'k'}.z_{ij'k'l}.C_{j',(k'-1)}$, then the linear form of this constraint is provided as constraints (34) and (35).

$$R_{ij'k'l} \geq \left(x_{ij'k'}.z_{ij'k'l} - 1\right).M + C_{j',(k'-1)} \quad for\ i = 1,2,\dots,m\ and\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'}\ and\ l = 1,2,\dots,l_i \tag{34}$$

$$R_{ij'k'l} \geq x_{ij'k'}.z_{ij'k'l}.(-M) \quad for\ i = 1,2,\dots,m\ and\ j' = 1,2,\dots,n\ and\ k' = 1,2,\dots,k'_{j'}\ and\ l = 1,2,\dots,l_i \tag{35}$$

Equation (31) is also a non-linear equation where two binary variables are multiplied. If assumed $U_{ijkj'k'l} = z_{ijk,(l-1)}.z_{ij'k'l}.S_{ijkj'k'}$, then this non-linear equation is converted into a linear equation form through constraints (36) and (37).

$$U_{ijkj'k'l} \geq \frac{1}{2} \times S_{ijkj'k'} \times \left(z_{ijk,(l-1)} + z_{ij'k'l}\right) + M \times \left(z_{ijk,(l-1)} + z_{ij'k'l} - 2\right) \quad for\ i = 1,2,\ldots,m\ and\ j = 0,1,\ldots,n\ and\ k = 0,1,\ldots,k_j\ and\ j' = 1,2,\ldots,n\ and\ k' = 1,2,\ldots,k'_{j'}\ and\ l = 1,2,\ldots,l_i \tag{36}$$

$$U_{ijkj'k'l} \geq 0 \quad for\ i = 1,2,\ldots,m\ and\ l = 1,2,\ldots,l_i \tag{37}$$

4- Numerical examples

In this section, a set of numerical data are examined by using mathematical optimization software, in order to evaluate the flexibility and efficiency of the model. In single-objective problems must be determined the best solution, while multi-objective problems may achieve no optimal solution for all objectives. Therefore, for multi-objective optimization problems, a set of Pareto solutions (non-dominant) should be determined. There are a variety of methods to solve multi-objective optimization problems; in this research, the ε-constraint method is used. Some strengths of this approach are: 1) Changes in the ε value result in different optimum solutions; 2) Different scaling of objective functions shows no adverse effect; and, 3) This approach is also suitable for non-convex problems.

In general, two numerical examples are provided where the relevant data are generated in logical and random fashion. In the small- and large-scales, these examples reveal the well performance of the developed model for JSP. The below discusses each of the numerical examples.

4-1- Validation of the model with a small-sized numerical example

In this example, three different operations of three jobs need to be performed by three machines. The aim is to determine the optimal sequence of three operations for each job, so that both makespan criterion and maximum tardiness are simultaneously minimized. Table 2 provides data on the duration of each operation of the jobs on certain machines. The delivery times are also given for the jobs. $O_{j,k}$ shows the k^{th} operation of the j^{th} job.

Table 2. Duration of operations on machines, and delivery times for each job, for the first example (per unit time)

	d_j		M_1	M_2	M_3
Job 1	12	$O_{1,1}$	3	-	-
		$O_{1,2}$	-	4	-
		$O_{1,3}$	-	-	4
Job 2	15	$O_{2,1}$	-	-	2
		$O_{2,2}$	3	-	-
		$O_{2,3}$	-	4	-
Job 3	18	$O_{3,1}$	5	-	-
		$O_{3,2}$	-	-	5
		$O_{3,3}$	-	2	-

Table 3 shows the sequence-dependent setup times. For example, the amount of 1 for the element 100 and the sub-element 11 means that the amount of setup time will be equal to 1 unit of time, if Operation 1 of Job 1 is placed on Machine 1, following the 0^{th} dummy operations of the 0^{th} dummy job. Since, for example, it is not feasible to place Operation 1 of Job 1 on the machines after the process of Operation 2 of Job 1 on each machines, no values could be assigned to the corresponding element.

Table 3. Sequence-dependent setup times on machines (per unit time)

ijk \ j'k'	11	12	13	21	22	23	31	32	33
100	1	2	1	1	0	0	1	3	4
111	-	2	2	1	4	1	0	1	3
112	-	-	1	2	3	4	1	2	2
113	-	-	-	1	3	2	0	2	1
121	2	0	1	-	4	4	2	1	2
122	1	2	3	-	-	2	1	0	4
123	2	4	1	-	-	-	1	2	0
131	1	2	3	1	3	1	-	3	1
132	3	1	2	4	2	1	-	-	2
133	1	3	2	2	0	1	-	-	-
200	2	3	0	4	1	2	0	2	3
211	-	1	2	5	3	1	0	1	2
212	-	-	1	2	3	2	2	1	0
213	-	-	-	2	1	3	4	0	1
221	1	0	2	-	3	1	1	2	2
222	2	1	1	-	-	2	2	3	4
223	1	2	1	-	-	-	1	3	0
231	2	1	1	2	4	2	-	0	1
232	2	3	2	2	1	0	-	-	2
233	1	2	4	3	0	2	-	-	-
300	1	4	2	2	0	1	2	1	3
311	-	2	1	1	2	2	5	2	3
312	-	-	1	1	2	3	3	4	4
313	-	-	-	2	1	3	4	0	1
321	1	2	2	-	3	1	2	4	0
322	1	0	1	-	-	3	2	2	2
323	1	3	4	-	-	-	1	3	0
331	3	1	1	2	0	2	-	0	1
332	3	3	2	0	1	1	-	-	3
333	0	1	2	4	5	2	-	-	-

Using the sequence-dependent setup times and solving the two-objective JSP model with mathematical optimization software, figure 1 is achieved. Here, the optimal solution has just one single point because of the small size of the problem. As seen from figure 1, the optimal amount of the makespan criterion is equal to 24 units of time. Moreover, given the tardiness and completion times of each job, the optimal amount of the maximum tardiness will be equal to 6 units of time. In this example, jobs 2 and 3 show the maximum tardiness.

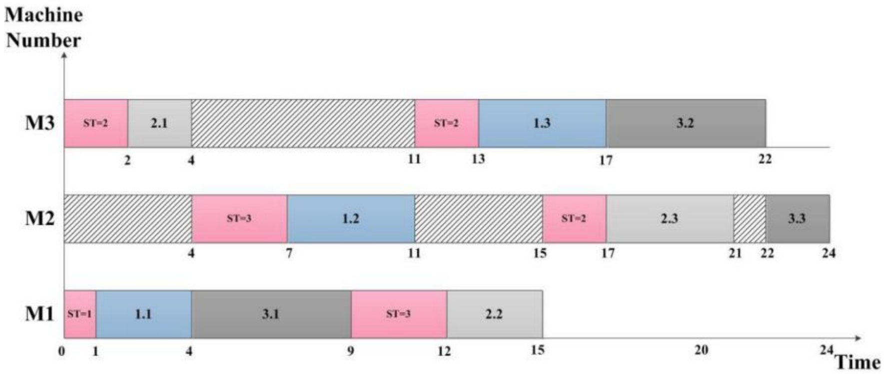

Fig 1. Results of the first numerical example solved with two objective functions

4-2- Validation of the model with a relatively larger numerical example

In this example, different operations of five jobs must be performed by three machines. The aim is to determine the optimal sequence for each job, so that both makespan criterion and maximum tardiness are minimized at the same time. Table 4 shows data related to the duration of each operation of the jobs on certain machines. The delivery times are also given for the jobs. $O_{j,k}$ shows the k^{th} operation of the j^{th} job.

Table 4. Duration of operations on machines, and delivery times for each job, for the second example (per unit time)

	d_j		M_1	M_2	M_3
Job 1	12	$O_{1,1}$	-	-	6
Job 2	15	$O_{2,1}$	7	-	-
		$O_{2,2}$	-	3	-
Job 3	18	$O_{3,1}$	-	1	-
		$O_{3,2}$	-	-	2
		$O_{3,3}$	-	-	4
Job 4	25	$O_{4,1}$	-	5	-
		$O_{4,2}$	-	-	3
		$O_{4,3}$	-	2	-
		$O_{4,4}$	1	-	-
Job 5	38	$O_{5,1}$	1	-	-
		$O_{5,2}$	-	-	3
		$O_{5,3}$	5	-	-
		$O_{5,4}$	-	6	-
		$O_{5,5}$	2	-	-

Due to the high volume data related to the sequence-dependent setup times, from mentioning the amount these times are ignored. Considering the two-objective JSP, and solving the example by mathematical optimization software, a series of Pareto optimal solutions is achieved as represented by figure 2. The optimum solutions include all points on the curve.

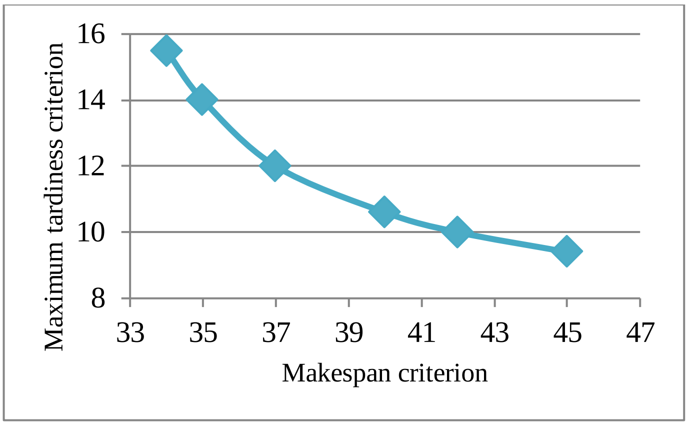

Fig 2. A set of Pareto optimal solutions

One of the Pareto solutions results in a sequence of job operations as figure 3. In an optimal situation, as found, the amount of the makespan criterion is equal to 37 units of time. Further, given the tardiness and completion times of each job, the optimal amount of the maximum tardiness will be equal to 12 units of time. In the present example, job 4 shows the maximum tardiness time.

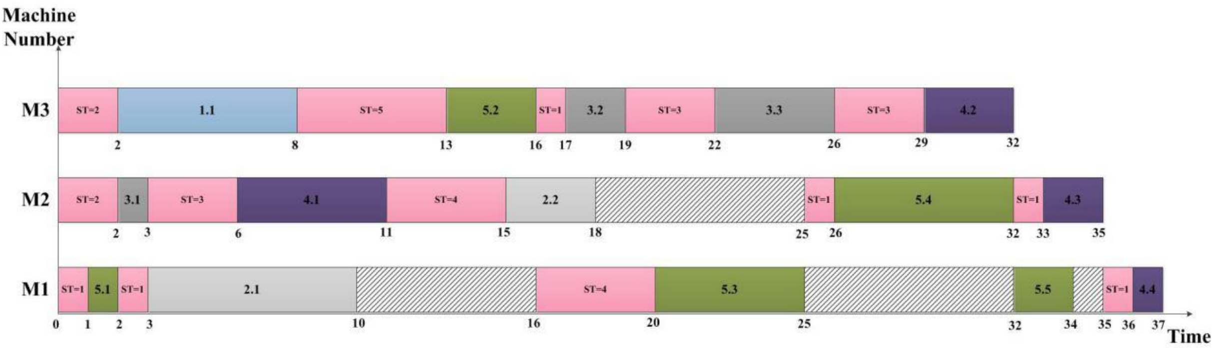

Fig 3. One of the Pareto solutions from the second numerical example solved with two-objective functions

Now, the implementation of the problem model and the examination of each objective function separately give us interesting results as to compare with the two-objective scenario.

If the production management aims only to optimize the makespan criterion, then figure 4 is achieved.

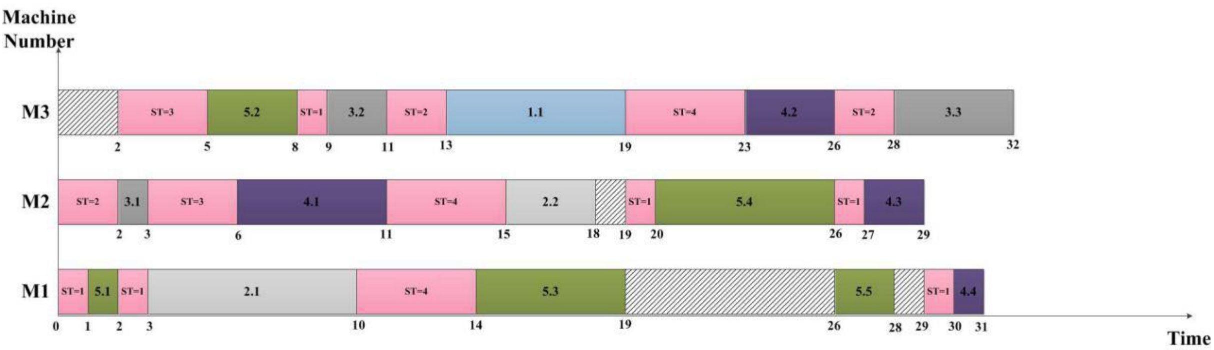

Fig 4. Results of the second numerical example solved with only the makespan criterion

As seen from figure 4, the optimal amount of the makespan criterion is equal to 32 units of time. Further, given the tardiness and completion times of each job, the non-optimal amount of the maximum tardiness will be equal to 14 units of time. In here, job 3 presents the maximum tardiness.

When the production management aims only to optimize the maximum tardiness, figure 5 is obtained.

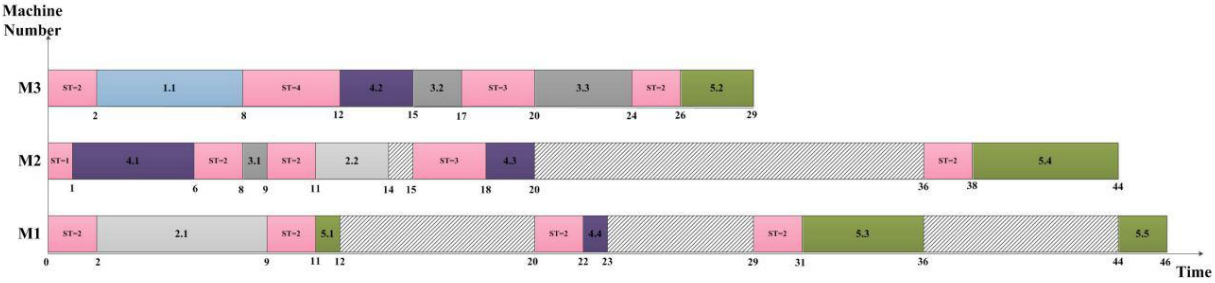

Fig 5. Results of the second numerical example solved with only the maximum tardiness criterion

According to the tardiness and completion times of each job, figure 5 illustrates that the optimal amount of the maximum tardiness is equal to 8 units of time. Here, Job 5 finds the maximum tardiness. Also, the non-optimal amount of the makespan criterion will be equal to 46 units of time.

4-3- Sensitivity analysis on parameters

To investigate the effect of changes in the model parameters on the objective functions, the values of two parameters of P_{jk} and $S_{ijkj'k'}$ are simultaneously changed. Table 5 presents the results.

Table 5. The effect of changes in parameters on objective functions

$S_{ijkj'k'}$						
-10%		0%		+10%		
$C_{\max}$	$T_{\max}$	$C_{\max}$	$T_{\max}$	$C_{\max}$	$T_{\max}$	P_{jk}
27.65	11.65	30.9	13.55	34.85	15.1	-5%
28.5	12.5	32	14	35.3	15.3	0%
29.45	16.25	33.05	18.25	36.75	19.95	+5%

Figure 6 shows the results of changes in two parameters on the makespan criterion. The makespan criterion is obviously increased if the duration of the operation (P_{jk}) is fixed and the setup times ($S_{ijkj'k'}$) is increased. Moreover, the fixed setup times and the reduced operation duration lead to a reduced makespan criterion.

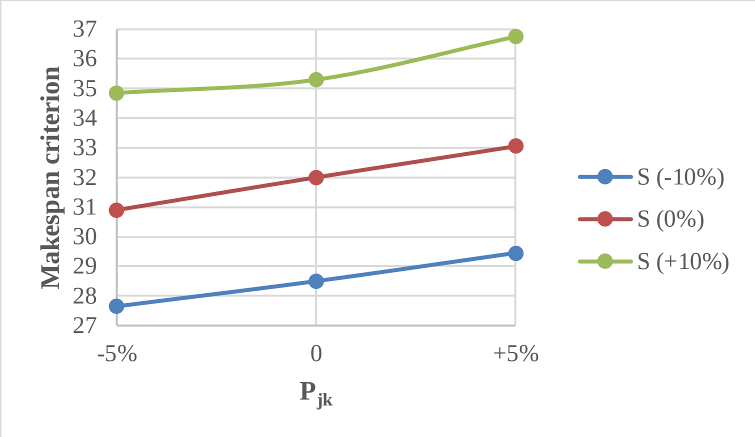

Fig 6. Sensitivity analysis of parameters and their impact on makespan criterion under two-objective mode

Figure 7 shows the results of changes in two parameters on the maximum tardiness criterion. The fixed duration of the operation (P_{jk}) and the increased setup times ($S_{ijkj'k'}$) result in an increasing in the maximum tardiness. Also, when the setup times are fixed and the operation duration is reduced, then the maximum tardiness is reduced.

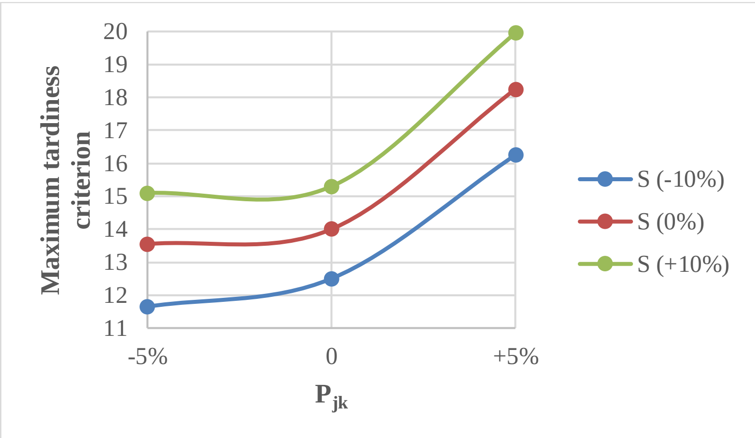

Fig 7. Sensitivity analysis of parameters and their influence on the maximum tardiness criterion under two-objective mode

5- Conclusion and future researches

This research develops a model for solving a two-objective job shop scheduling problem (JSP) with sequence-dependent setup times (SDST). The first objective function is to minimize the makespan criterion which has been widely studied by a wide range of research works. The second objective function is to minimize the maximum tardiness time; however, it appears to have been less investigated. Considering SDST and optimizing both objectives simultaneously make us closer to real-world scheduling problems and lead to more accurate results. The MINLP model was converted into a MILP model. The ε-constraint method was used to solve the mentioned model, and a set of Pareto optimal solutions were obtained. Further, a set of numerical data were investigated to represent the model efficiency and flexibility by using mathematical optimization software. The computational results clearly revealed the model efficiency, whereas the sensitivity analysis and the checking of changes in both objectives confirmed the high flexibility of the model. The expanded model and the results can be efficiently used to solve JSPs in the real world, especially for manufacturing companies with having setup times and delivery time constraints.

For this research, the first and most important innovation element includes the considering of sequence-dependent setup times for modeling the scheduling problems, in addition to job completion times. This makes the research works close to the real-world manufacturing systems. The second element is the application of a two-objective approach for problem modeling and solution, so both criteria – makespan and maximum tardiness – are simultaneously minimized. In order to expand the present model for future studies, some recommendations are suggested. First, the development of an efficient algorithm can be useful for solving the model with large-scale data. Second, the duration of the operation for each job and also the setup times can be regarded as non-deterministic values in order to achieve the real-world issues. Third, it will be desirable to consider different objectives such as minimizing the total completion time of operations or minimizing the total tardiness times.

References

Al-Hinai, N. and T. Y. ElMekkawy (2011a). An efficient hybridized genetic algorithm architecture for the flexible job shop scheduling problem. Flexible Services and Manufacturing Journal **23**(1): 64-85.

Al-Hinai, N. and T. Y. ElMekkawy (2011b). Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. International Journal of Production Economics **132**(2): 279-291.

Asano, M. and H. Ohta (2002). A heuristic for job shop scheduling to minimize total weighted tardiness. Computers & Industrial Engineering **42**(2–4): 137-147.

Bagheri, A. and M. Zandieh (2011). Bi-criteria flexible job-shop scheduling with sequence-dependent setup times—variable neighborhood search approach. Journal of Manufacturing Systems **30**(1): 8-15.

Brucker, P., et al. (1994). A branch and bound algorithm for the job-shop scheduling problem. Discrete Applied Mathematics **49**(1–3): 107-127.

Carlier, J. and E. Pinson (1989). An Algorithm for Solving the Job-Shop Problem. Management Science **35**(2): 164-176.

Della Croce, F., et al. (1995). A genetic algorithm for the job shop problem. Computers & Operations Research **22**(1): 15-24.

Ebadi, A. and G. Moslehi (2013). An optimal method for the preemptive job shop scheduling problem. Computers & Operations Research **40**(5): 1314-1327.

Fattahi, P. and F. Daneshamooz (2017). Hybrid algorithms for Job shop Scheduling Problem with Lot streaming and A Parallel Assembly Stage. Journal of Industrial and Systems Engineering **10**(3): 92-112.

Fattahi, P., et al. (2007). Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. Journal of Intelligent Manufacturing **18**(3): 331-342.

Garey, M. R., et al. (1976). The Complexity of Flowshop and Jobshop Scheduling. Mathematics of Operations Research **1**(2): 117-129.

Gonçalves, J. F., et al. (2005). A hybrid genetic algorithm for the job shop scheduling problem. European Journal of Operational Research **167**(1): 77-95.

González, M. A., et al. (2013). Lateness minimization with Tabu search for job shop scheduling problem with sequence dependent setup times. Journal of Intelligent Manufacturing **24**(4): 741-754.

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

Kuhpfahl, J. and C. Bierwirth (2016). A study on local search neighborhoods for the job shop scheduling problem with total weighted tardiness objective. Computers & Operations Research **66**: 44-57.

Kurdi, M. (2016). An effective new island model genetic algorithm for job shop scheduling problem. Computers & Operations Research **67**: 132-142.

Laarhoven, P. J. M. v., et al. (1992). Job Shop Scheduling by Simulated Annealing. Operations Research **40**(1): 113-125.

Lee, Y. H. and M. Pinedo (1997). Scheduling jobs on parallel machines with sequence-dependent setup times. European Journal of Operational Research **100**(3): 464-474.

Li, J.-Q., et al. (2014). A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities. Applied Mathematical Modelling **38**(3): 1111-1132.

Mattfeld, D. C. and C. Bierwirth (2004). An efficient genetic algorithm for job shop scheduling with tardiness objectives. European Journal of Operational Research **155**(3): 616-630.

Mousakhani, M. (2013). Sequence-dependent setup time flexible job shop scheduling problem to minimise total tardiness. International journal of production research **51**(12): 3476-3487.

Naderi, B. and A. Azab (2014). Modeling and heuristics for scheduling of distributed job shops. Expert Systems with Applications **41**(17): 7754-7763.

Naderi, B., et al. (2009). Scheduling job shop problems with sequence-dependent setup times. International journal of production research **47**(21): 5959-5976.

Naderi, B., et al. (2009). Scheduling sequence-dependent setup time job shops with preventive maintenance. The International Journal of Advanced Manufacturing Technology **43**(1): 170-181.

Nowicki, E. and C. Smutnicki (1996). A Fast Taboo Search Algorithm for the Job Shop Problem. Management Science **42**(6): 797-813.

Özgüven, C., et al. (2012). Mixed integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence dependent setup times. Applied Mathematical Modelling **36**(2): 846-858.

Park, B. J., et al. (2003). A hybrid genetic algorithm for the job shop scheduling problems. Computers & Industrial Engineering **45**(4): 597-613.

Petrovic, S., et al. (2008). Fuzzy job shop scheduling with lot-sizing. Annals of Operations Research **159**(1): 275-292.

Pinedo (2012). Scheduling: theory, algorithms, and systems, Springer Science & Business Media.

Pinedo and M. Singer (1999). A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. Naval Research Logistics **46**(1): 1-17.

Ponnambalam, G. S., et al. (2000). A Tabu Search Algorithm for Job Shop Scheduling. The International Journal of Advanced Manufacturing Technology **16**(10): 765-771.

Qi, G. J., et al. (2000). The Application of Parallel Multipopulation Genetic Algorithms to Dynamic Job-Shop Scheduling. The International Journal of Advanced Manufacturing Technology **16**(8): 609-615.

Saidi-Mehrabad, M. and P. Fattahi (2007). Flexible job shop scheduling with tabu search algorithms. The International Journal of Advanced Manufacturing Technology **32**(5): 563-570.

Sharma, P. and A. Jain (2015). Performance analysis of dispatching rules in a stochastic dynamic job shop manufacturing system with sequence-dependent setup times: Simulation approach. CIRP Journal of Manufacturing Science and Technology **10**: 110-119.

Shen, L., et al. (2017). Solving the Flexible Job Shop Scheduling Problem with Sequence-Dependent Setup Times. European Journal of Operational Research.

Sun, X. and J. S. Noble (1999). An approach to job shop scheduling with sequence-dependent setups. Journal of Manufacturing Systems **18**(6): 416-430.

Taillard, E. D. (1994). Parallel Taboo Search Techniques for the Job Shop Scheduling Problem. ORSA Journal on Computing **6**(2): 108-117.

Tan, et al. (2015). Configuration and the advantages of the shifting bottleneck procedure for optimizing the job shop total weighted tardiness scheduling problem. Journal of scheduling: 1-24.

Watanabe, M., et al. (2005). A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem. Computers & Industrial Engineering **48**(4): 743-752.

Yang, S., et al. (2010). An improved constraint satisfaction adaptive neural network for job-shop scheduling. Journal of scheduling **13**(1): 17-38.

Zhang, et al. (2007). A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. Computers & Operations Research **34**(11): 3229-3242.

Zhang, et al. (2008). A very fast TS/SA algorithm for the job shop scheduling problem. Computers & Operations Research **35**(1): 282-294.

Zhang, et al. (2012). A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times. Computers & Operations Research **39**(7): 1713-1723.