# Unrelated Parallel Machines Scheduling Problem with Sequence Dependent Setup Times

**Vahid Kayvanfar, Amin Aalaei, Mahtab Hosseininia and Mahdi Rajabi**
**Department of Industrial Engineering**
**Amirkabir University of Technology**
**424 Hafez Ave., 15916-34311**
**Tehran, Iran**

## Abstract

A multi-criteria scheduling problem with the goal of minimizing the maximum completion time, so called makespan as well as earliness and tardiness penalties simultaneously on unrelated parallel machines is studied in this research in which jobs are sequence dependent setup times (SDST) and due dates are distinct. Jobs processing cost/time on different machines may vary, i.e., each machine can process each job at different processing time with respect to the other machines. No jobs preemption is allowed and no inserted idle time could be inserted into the schedule, after starting the process by machine. In the considered problem, a mathematical model is proposed which could be solved optimally by lingo in small size problems. In addition, a heuristic called Initial Sequence based on Earliness-Tardiness criterion on Parallel machines (ISETP) is presented so as to acquire the jobs sequence on parallel machines regarding minimizing total weighted tardiness and earliness. Computational results demonstrate that the proposed technique is a reliable one which can solve such complicated problems within very intangible computational time.

## Keywords
Just-in-Time (JIT); Unrelated parallel machines; Makespan; Sequence dependent setup time (SDST).

## 1. Introduction

Owing to the remarkable increase in the competitive productive world in the last two decades, JIT production has proved to be an essential requirement of world class manufacturing. Identifying and eliminating waste components during the production process such as waiting time, transportation, inventory, movement and defective products is included in JIT philosophy (Wang, 2006). Since, neither earliness nor tardiness is desirable, because of one can represent manufacturer concerns and the other one may represent the customer concerns, we study parallel processors environment in which both earliness and tardiness as well as makespan are minimized, however most of the models which have surveyed both earliness and tardiness concurrently are single machine ones. On the other hand, makespan is one of the most widely studied objectives in the literatures and reducing it is of great interest. (Eswaramurthy, 2008).

Generally speaking, machine scheduling falls into two main categories, single machine and multi-machine problems. Although, single machines are easier to solve, they hardly occur in real manufacturing systems. On the other hand, most scheduling problems for parallel machines have real occurrences in industrial systems. Scheduling of unrelated parallel machines is one of the most important and yet complicated subjects in the multi-machine manufacturing environments. Despite a large number of researches are done on parallel machines, but few of them have surveyed unrelated parallel ones or sequence-dependent setups.

There are several papers in which earliness and tardiness criteria are studied simultaneously on parallel machines. Of them, Ventura and Kim (2003) considered the problem of scheduling jobs on parallel machines where jobs have different due dates and may require, besides machines, certain additional limited resources for their handling and processing. Toksari and Guner (2009) considered a parallel machine earliness/tardiness scheduling problem with different penalties under the effects of position based learning and linear and nonlinear deterioration. As indicated by Morton and Pentico (1993) and Liaw et al. (2003) due-date-related problems for multi-machine environments are usually computationally complex and hence most existing results are typically for problems with small sizes or simple settings. Cheng and Sin (1990) studied a comprehensive review on parallel machine scheduling problems with conventional performance measures based on due date, completion time, and flow time. With respect to minimizing the total weighted tardiness in unrelated parallel machines scheduling problem, Liaw et al. (2003) proposed a two-phase heuristic for solving the job scheduling problem. Bank and Werner (2001) considered unrelated parallel machine regarding release date as well as common due date. Rocha et al. (2008) studied unrelated parallel machines considering sequence and machine-dependent setup times, due dates and weighted jobs. Of multi-criteria researches, one could refer to Gurel and Akturk (2007) which proposed an improved branch and bound algorithm to solve a bi-criteria allocation and processing time problem for non-identical parallel CNC machines. Vallada and Ruiz (2011) presented a genetic algorithm for the unrelated parallel machine scheduling problem considering machine and job sequence dependent setup times. Lin et al. (2011) performed a study in which the performance of various heuristics is compared with one meta-heuristic for unrelated parallel machine scheduling problems. Their objective was to minimize makespan, total weighted completion time and total weighted tardiness. M'Hallah and Al-Khamis (2012) studied minimum weighted earliness-tardiness parallel machine scheduling problem with distinct deterministic known due dates regarding allowable machine idle time. Janiak et al. (2013) studied problems of scheduling *n* jobs on *m* identical parallel machines in which a common due window had to be assigned to all jobs. Their objective was to find a job schedule as well as location and size of the due window such that a sum of costs associated with job earliness, job tardiness and due window location and size is minimized. Recently, Kayvanfar et al. (2014) addressed minimizing total weighted tardiness and earliness on unrelated parallel machines in which the jobs processing times are controllable. In fact, it is assumed that the jobs processing times can vary within a given interval, i.e. it's permitted to compress or expand in return for compression/expansion cost.

## 2. Problem formulation

A Mixed Integer Programming (MIP) mathematical model is proposed to minimize total earliness and tardiness in addition to makespan, simultaneously, on unrelated parallel machines considering SDST assumption. In this context, a set of *N* jobs denoted by 1, 2,.., *n* has to be processed on a set of *M* unrelated parallel machines denoted by $M_1$, $M_2$, …, $M_m$. The setups are assumed to be simultaneously machine and job dependent, consequently a setup time $S_{kim}$ is incurred, when a given job *k* is processed immediately after job *i* on machine *j*.

### 2.1 Assumptions

Several assumptions are assumed in this study which are stated as follows:

- All jobs and machines are available in time zero.
- The setup time for each job on each machine is sequence-dependent.
- The process time of each job on each machine differs of each other.
- After starting the process by machine, no idle time can inserted into the schedule.
- Each job has a distinct due date and must be processed only one time.
- No preemption of operations of each job is allowed.
- Machines are available throughout the scheduling period (i.e., no breakdown).
- Number of jobs and machines are fixed.
- Each machine can process only one operation at a time.
- All machines are unrelated and each job can be processed by a free machine.

### 2.2 Notations

2.2.1. Subscripts

| | |
|---|---|
| $N$ | Number of jobs |
| $M$ | Number of machines |
| $i, k$ | Index for job ($i,k$ =1,2,…,$N$) |
| $j$ | Index for priorities ($j$=1, 2,…,$J$) |
| $m$ | Index for machine ($m$=1, 2,…,$M$) |

2.2.2. Input parameters

| | |
|---|---|
| $p_{im}$ | Processing time of job *i* on machine *m* |

$\alpha_i$ The earliness unit penalty of job *i*

$\beta_i$ The tardiness unit penalty of job *i*

$d_i$ Due date of job *i*

$\lambda$ Factory costs per time unit (including machines, labor and variable production costs and the costs dependent to the work time)

$S_{kim}$ Setup time for assigning job *i* after job *k* on machine *m*

2.2.3. Decision variables

$C_i$ Completion time of job *i*

$C_{max}$ Total completion time or makespan

$E_i$ Earliness of job *i*; $E_i = \max\{0, d_i - C_i\}$

$T_i$ Tardiness of job *i*; $T_i = \max\{0, C_i - d_i\}$

$y_{ijm}$ 1 if job *i* on machine *m* in priority *j*; otherwise, it is zero.

**2.3 The mathematical model**

$$\min\ Z = \min\left(\lambda C_{\max} + \sum_{i=1}^{N}\left(\alpha_i E_i + \beta_i T_i\right)\right) \tag{1}$$

$$\sum_{m=1}^{M}\sum_{j=1}^{J} y_{ijm} = 1 \qquad \forall i\ ; \tag{2}$$

$$\sum_{i=1}^{N} y_{ijm} \le 1 \qquad \forall j, m\ ; \tag{3}$$

$$C_i - d_i \le T_i \qquad \forall i\ ; \tag{4}$$

$$d_i - C_i \le E_i \qquad \forall i\ ; \tag{5}$$

$$y_{i1m} \times p_{im} \le C_i \qquad \forall i, m; \tag{6}$$

$$\left(\sum_{m=1}^{M}\sum_{j\ge 2}^{J}\sum_{k\ne i}^{N} y_{kj-1m}.y_{ijm}(C_k + S_{kim})\right) + \sum_{m=1}^{M}\sum_{j=1}^{J} p_{im}.y_{ijm} = C_i \qquad \forall i\ ; \tag{7}$$

$$C_{\max} \ge C_i \qquad \forall i\ ; \tag{8}$$

$$y_{ijm} \in \{0,1\} \qquad \forall i, j, m\ ; \tag{9}$$

$$T_i, E_i, C_i \ge 0 \qquad \forall i\ ; \tag{10}$$

Equation (1) is our objective function which aims minimizing total weighted earliness and tardiness as well as maximum completion time or makespan. Equality (2) ensures that job *i* in only one priority *j* and on only one machine could be processed. Inequality (3) guarantees that in priority *j* on machine *m* only one job could be processed. Constraints (4) and (5) define the earliness and tardiness of job *i* which both these constraints must be minimized. Constraints (6) and (7) together ensure that only after starting the process by machine, no idle time could be inserted into the schedule, and no job preemption is allowed. Set (8) defines the maximum completion time which obviously is as large as all completion time on each machine. Finally, Set (9) defines the binary variables and Set (10) identifies non-negativity of decision variables.

### 2.4. Linearization of the proposed model

In this section, an attempt is made to linearize the mathematical model proposed in previous section.

#### 2.4.1. Procedure

The linearization procedure that we propose here consists of two steps that are given by the two propositions stated below. The non-linear terms in the Constraint (7) is multiplication of binary and integer variables which can be linearized using the following auxiliary variables $F_{ikjm}$ and $G_{ikjm}$. Each proposition for linearization is followed by a proof that illustrates the meaning of each auxiliary (linearization) variable and the expressions where they are used.

**Proposition 1.** The non-linear terms in the Constraint (7) of the mathematical model can be linearized with $F_{ikjm} = y_{kj-1m} \, . \, y_{ijm}$, under the following sets of constraints:

$$F_{ikjm} \geq y_{kj-1m} + y_{ijm} - 1.5 \qquad \forall j \geq 2, i \neq k, m; \qquad (11)$$

$$1.5 \times F_{ikjm} \leq y_{kj-1m} + y_{ijm} \qquad \forall j \geq 2, i \neq k, m; \qquad (12)$$

**Proof.** Consider the following two cases:

(i) $y_{kj-1m} \, . \, y_{ijm} = 1. \qquad \forall j \geq 2, i \neq k, m;$

Such a situation arises when $y_{kj-1m} = y_{ijm} = 1$. So, Constraint (11) implies $F_{ikjm} \geq 0.5$ which ensures that $F_{ikjm} = 1$.

(ii) $y_{kj-1m} \, . \, y_{ijm} = 0$. Such a situation arises under one of the following three cases:

(a) $y_{kj-1m} = 1$ and $y_{ijm} = 0. \qquad \forall j \geq 2, i \neq k, m;$
(b) $y_{kj-1m} = 0$ and $y_{ijm} = 1. \qquad \forall j \geq 2, i \neq k, m;$
(c) $y_{kj-1m} = 0$ and $y_{ijm} = 0. \qquad \forall j \geq 2, i \neq k, m;$

In all of these cases, the value of $F_{ikjm} = 0$, because in these cases, Constraint (12) implies $1.5\times F_{ikjm} \leq 0$ or 1 and so ensures that $F_{ikjm} = 0$. Since $F_{ikjm}$ has not a strictly positive cost coefficient, the minimizing objective function doesn't ensure that $F_{ikjm} = 0$. Thus, Constraint (12) should be added to the mathematical model.

**Proposition 2.** Also the non-linear Constraint (7) could be linearized by the following transformation $G_{ikjm} = F_{ikjm} \, .C_i$, under the following sets of constraints:

$$G_{ikjm} \leq C_i + A(1 - F_{ikjm}) \qquad \forall j \geq 2, i \neq k, m; \qquad (13)$$

$$G_{ikjm} \geq C_i - A(1 - F_{ikjm}) \qquad \forall j \geq 2, i \neq k, m; \qquad (14)$$

$$G_{ikjm} \leq A.F_{ikjm} \qquad \forall j \geq 2, i \neq k, m; \qquad (15)$$

**Proof.** Consider the following two sections:
This section can be shown for each of the two possible cases that can arise.

(i) $F_{ikjm} \, . \, C_i = C_i \qquad \forall j \geq 2, i \neq k, m;$

Such a situation arises when $F_{ikjm} = 1$ so, Constraints (13) and (14) implies $G_{ikjm} \leq C_i$ and $G_{ikjm} \geq C_i$ and ensures that $G_{ikjm} = C_i$.

(ii) $F_{ikjm} \, . \, C_i = 0$. Such a situation arises under one of the following three sub-cases:

(a) $F_{ikjm} = 1$ and $C_i = 0$. $\forall j \geq 2, i \neq k, m;$

(b) $F_{ikjm} = 0$ and $C_i > 0$. $\forall j \geq 2, i \neq k, m;$

(c) $F_{ikjm} = 0$ and $C_i = 0$. $\forall j \geq 2, i \neq k, m;$

In all of the three sub-cases given above, $G_{ikjm}$ takes the value of 0, because in these cases, Constraint (15) implies $G_{ikjm} \leq 0$ and ensures that $G_{ikjm} = 0$. Because $G_{ikjm}$ has not a strictly positive cost coefficient, the minimizing objective function doesn't ensures that $G_{ikjm} = 0$. Thus, Constraint (15) should be added to the mathematical model.

#### 2.4.2. The linearized model

We now present the linear mathematical model as follows:

*Min Z* = Eq. (1)

Subject to constraints (2) – (6) and (8) - (16)

$$G_{ikjm} \geq 0 \quad and \quad F_{ikjm} \in \{0,1\} \qquad \forall j \geq 2, i \neq k, m; \tag{16}$$

## 3. Initial Sequence based on Earliness-Tardiness criterion on Parallel machine

Since in this paper we aim total tardiness and earliness criteria simultaneously in context of JIT approach, our heuristic must satisfy such a criterion. So, in this section, a heuristic, called Initial Sequence based on Earliness-Tardiness criterion on Parallel machine (ISETP) is proposed to assign the jobs on the unrelated parallel machines which is trying to minimize the mentioned objective function.

### *ISETP*:

1. Sort jobs according to Earliest Due Date (EDD) criterion and put them in *unscheduled* jobs category, *US*.
2. Assign the first job to the first machine and set it in *scheduled* job category, *S*.

Do the following steps until no outstanding job is found:

3. Assign next *unscheduled* job in EDD order to each machine, separately and then calculate the $C_{max}$ of each machine.
4. Compute difference between due date of this job and $C_{max}$ of such machine ($C_{\max_j}$) called $U_j = \left| C_{\max_j} - d_j \right|$.
5. Select the assignment which has resulted in the minimum $U_j$ and assign that job to such a machine. If two or more values are equal, select the machine with minimum index (however there's no difference in selecting each of them).
6. Transfer this job from *US* category into *scheduled* one, *S*.
7. Update $C_{max}$ of each machine and go to Step 3.

In order to illustrate the ISETP performance, solving an example can be useful. It must be mentioned that in the following example processing time of all jobs are considered the same on different machines so as to simplify exemplifying the performance of the algorithm.

***Example 1***. Consider the following problem with 8 jobs and 3 parallel machines.

Table 1. Input data for an eight job problem with three machines

| $J$ | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|
| $P_j$ | 4 | 6 | 5 | 7 | 5 | 6 | 4 | 6 |
| $d_j$ | 5 | 5 | 11 | 6 | 13 | 13 | 11 | 20 |
| $\alpha_j$ | 0.5 | 1 | 1 | 1.25 | 1.5 | 1 | 1.5 | 0.5 |
| $\beta_j$ | 0.5 | 0.5 | 1.25 | 0.5 | 0.5 | 1 | 3 | 0.5 |

**Iteration #1:**

Step 1. Sort jobs according EDD rule and set them in *US* = {$J_1$, $J_2$, $J_4$, $J_3$, $J_7$, $J_5$, $J_6$, $J_8$}.

Step 2. Assign the first job to the first machine and set it in *scheduled* job category, *S* = {$J_1$}.

Step 3. Assign the second job ($J_2$) to each machine separately,

$C_{\max_1} = 4 + 6 = 10$, $C_{\max_2} = C_{\max_3} = 6$

Step 4. Compute all $U_j$ as follows:

$U_1 = \left|C_{\max_1} - d_1\right| = |10 - 5| = 5,\ U_2 = \left|C_{\max_2} - d_2\right| = |6 - 5| = 1,\ U_3 = \left|C_{\max_3} - d_3\right| = |6 - 5| = 1.$

Step 5. Select the min $U_j$ and assign $J_2$ to such a machine. Since two values are equal, select the machine with minimum index, i.e., machine #2.

Step 6. Transfer this job from *US* category to *S* one, *US* = {$J_4$, $J_3$, $J_7$, $J_5$, $J_6$, $J_8$} and *S* = {$J_1$, $J_2$}.

Step 7. Update $C_{max}$ of each machine, $C_{\max_1} = 4,\ C_{\max_2} = 6,\ C_{\max_3} = 0.$

**Iteration #2:**

Step 3. Assign the third job ($J_4$) to each machine separately,

$C_{\max_1} = 4 + 7 = 11$, $C_{\max_2} = 6 + 7 = 13$, $C_{\max_3} = 7.$

Step 4. Compute all $U_j$ as follows:

$U_1 = \left|C_{\max_1} - d_1\right| = |11 - 6| = 5,\ U_2 = \left|C_{\max_2} - d_2\right| = |13 - 6| = 7,\ U_3 = \left|C_{\max_3} - d_3\right| = |7 - 6| = 1.$

Step 5. Select the min $U_j$ and assign $J_4$ to such machine, i.e., machine #3.

Step 6. Transfer this job from *US* category to *S* one, *US* = {$J_3$, $J_7$, $J_5$, $J_6$, $J_8$} and *S* = {$J_1$, $J_2$, $J_4$}.

Step 7. Update $C_{max}$ of each machine, $C_{\max_1} = 4,\ C_{\max_2} = 6,\ C_{\max_3} = 7.$

**Iteration #3:**

Step 3. Assign the forth job ($J_3$) to each machine separately,

$C_{\max_1} = 4 + 5 = 9$, $C_{\max_2} = 6 + 5 = 11$, $C_{\max_3} = 7 + 5 = 12.$

Step 4. Compute all $U_j$ as follows:

$U_1 = \left|C_{\max_1} - d_1\right| = |9 - 11| = 2,\ U_2 = \left|C_{\max_2} - d_2\right| = |11 - 11| = 0,\ U_3 = \left|C_{\max_3} - d_3\right| = |12 - 11| = 1.$

Step 5. Select the min $U_j$ and assign $J_3$ to this machine, i.e., machine #2.

Step 6. Transfer this job from *US* category to *S* one, *US* = {$J_7$, $J_5$, $J_6$, $J_8$} and *S* = {$J_1$, $J_2$, $J_4$, $J_3$}.

Step 7. Update $C_{max}$ of each machine, $C_{\max_1} = 4,\ C_{\max_2} = 11,\ C_{\max_3} = 7.$

**Iteration #4:**

Step 3. Assign the fifth job ($J_7$) to each machine separately,

$C_{\max_1} = 4+4 = 8$, $C_{\max_2} = 11+4 = 15$, $C_{\max_3} = 7+4 = 11$.

Step 4. Compute all $U_j$ as follows:

$U_1 = \left|C_{\max_1} - d_1\right| = |8-11| = 3,\ U_2 = \left|C_{\max_2} - d_2\right| = |15-11| = 4,\ U_3 = \left|C_{\max_3} - d_3\right| = |11-11| = 0.$

Step 5. Select the min $U_j$ and assign $J_7$ to this machine, i.e., machine #3.

Step 6. Transfer this job from *US* category to *S* one, *US* = {$J_5$, $J_6$, $J_8$} and *S* = {$J_1$, $J_2$, $J_4$, $J_3$, $J_7$}.

Step 7. Update $C_{max}$ of each machine, $C_{\max_1} = 4,\ C_{\max_2} = 11,\ C_{\max_3} = 11.$

**Iteration #5:**

Step 3. Assign the sixth job ($J_5$) to each machine separately,

$C_{\max_1} = 4+5 = 9$, $C_{\max_2} = 11+5 = 16$, $C_{\max_3} = 11+5 = 16$.

Step 4. Compute all $U_j$ as follows:

$U_1 = \left|C_{\max_1} - d_1\right| = |9-13| = 4,\ U_2 = \left|C_{\max_2} - d_2\right| = |16-13| = 3,\ U_3 = \left|C_{\max_3} - d_3\right| = |16-13| = 3.$

Step 5. Select the min $U_j$ and assign $J_5$ to this machine. Since two values are equal, select the machine with minimum index, i.e., machine #2.

Step 6. Transfer this job from *US* category to *S* one, *US* = {$J_6$, $J_8$} and *S* = {$J_1$, $J_2$, $J_4$, $J_3$, $J_7$, $J_5$}.

Step 7. Update $C_{max}$ of each machine, $C_{\max_1} = 4,\ C_{\max_2} = 16,\ C_{\max_3} = 11.$

**Iteration #6:**

Step 3. Assign the seventh job ($J_6$) to each machine separately,

$C_{\max_1} = 4+6 = 10$, $C_{\max_2} = 16+6 = 22$, $C_{\max_3} = 11+6 = 17$.

Step 4. Compute all $U_j$ as follows:

$U_1 = \left|C_{\max_1} - d_1\right| = |10-13| = 3,\ U_2 = \left|C_{\max_2} - d_2\right| = |22-13| = 9,\ U_3 = \left|C_{\max_3} - d_3\right| = |17-13| = 4.$

Step 5. Select the min $U_j$ and assign $J_6$ to this machine. i.e., machine #1.

Step 6. Transfer this job from *US* category to *S* one, *US* = {$J_8$} and *S* = {$J_1$, $J_2$, $J_4$, $J_3$, $J_7$, $J_5$, $J_6$}.

Step 7. Update $C_{max}$ of each machine, $C_{\max_1} = 10,\ C_{\max_2} = 16,\ C_{\max_3} = 11.$

**Iteration #7:**

Step 3. Finally, assign the last job in EDD order ($J_8$) to each machine separately,

$C_{\max_1} = 10+6 = 16$, $C_{\max_2} = 16+6 = 22$, $C_{\max_3} = 11+6 = 17$.

Step 4. Compute all $U_j$ as follows:

$U_1 = \left|C_{\max_1} - d_1\right| = |16-20| = 4,\ U_2 = \left|C_{\max_2} - d_2\right| = |22-20| = 2,\ U_3 = \left|C_{\max_3} - d_3\right| = |17-20| = 3.$

Step 5. Select the min $U_j$ and assign $J_8$ to this machine. i.e., machine #2.

Step 6. Transfer this job from *US* category to *S* one, *US* = { } and *S* = {$J_1$, $J_2$, $J_4$, $J_3$, $J_7$, $J_5$, $J_6$, $J_8$}.

Step 7. Update $C_{max}$ of each machine, $C_{\max_1} = 10,\ C_{\max_2} = 22,\ C_{\max_3} = 11.$

Since there's no outstanding job in *US* category, the algorithm is terminated. For the sake of comparison of obtained solution with optimum one, this problem is solved by lingo which the final solutions are as follows:

Table 2. Comparison of ISETP and Lingo in given example

| **ISETP** | **Lingo** |
|---|---|
| Z = 7 | $Z^* = 6$ |
| M (1) = $J_1 - J_6$ | M (1) = $J_2 - J_7$ |
| M (2) = $J_2 - J_3 - J_5 - J_8$ | M (2) = $J_1 - J_3 - J_5$ |
| M (3) = $J_4 - J_7$ | M (3) = $J_4 - J_6 - J_8$ |

As could be seen in Table 2, the obtained solution with the proposed ISETP is very close to the optimum one, however the sequence of the assigned jobs on each machine differs from the optimal one.

## 4. Numerical Experiences

A set of test examples are implemented on the SDST parallel machine scheduling problem. We run these instances on PC with a 2.66 GHz Intel Core 2 Duo processor and 4 GB RAM memory.

Table 3. Input parameters distribution

| **Input parameters** | **Distribution** |
|---|---|
| Processing time ( $p_i^j$ ) | ~ *DU*[30, 60] |
| Due dates ($d_j$) | ~ U[$d_{min}$, $d_{min}+\rho P$] |
| Earliness penalty ($\alpha_{ij}$) | ~ *U*(0.5, 2.5) |
| Tardiness penalty ($\beta_{ij}$) | ~ *U*(0.5, 2.5) |
| Setup times ($S_{kim}$) | ~ DU[10, 90] |
| Number of jobs ($n$) | 10 |
| Number of machines ($m$) | 2−3−5 |

Where $d_{min} = max\ (0,\ P(\upsilon - \rho/2))$ and $P = 1/m \sum_{j=1}^{m} \sum_{i=1}^{n} p_i^j$ . The expression of $P$ aims at satisfying the criteria of scale invariance and regularity described by Hall and Posner (2001) for generating experimental scheduling instances. The processing times are discretely uniformly distributed within range 30 and 60. Also, earliness and tardiness penalties ($\alpha_{ij}$ and $\beta_{ij}$) are both uniformly distributed as (0.5, 2.5). Setup times ($S_{kim}$) are generated from a discrete uniform distribution between 10 and 90. The two parameters $\upsilon$ and $\rho$ are the tardiness and range parameters, respectively. In this paper, we consider $\upsilon \in \{0.2, 0.5, 0.8\}$ and $\rho \in \{0.2, 0.5, 0.8\}$. For each triple ($m$, $\upsilon$, $\rho$) three instances are generated in which each case has run 5 times in all methods so as to guarantee constancy of these techniques. Consequently, considering $M$ = 2, 3 and 5, $N$ = 10, $\upsilon$ = 0.2, 0.5 and 0.8 and finally $\rho$ = 0.2, 0.5 and 0.8, for each combination of such triple we have $3^3 \times 3 \times 5 = 405$ problems totally.
Since these problems could be solved optimally via lingo, the percentage relative error (*PRE*) is used as the performance measure as follows:

$$PRE = \frac{A\lg_{sol} - O}{O} \times 100 \quad (17)$$

Where $Alg_{sol}$ is the objective value obtained by the proposed heuristic and $O$ is the optimum value obtained by lingo.

Table 4. Computational results for sample generated problems

| M | J | $\upsilon$ | $\rho$ | Lingo | ISETP | |
|---|---|---|---|---|---|---|
| | | | | MCPU Time | MCPU Time | $PRE_{avg}$ |
| 2 | 10 | 0.2 | 0.2 | 8469.82 | 0.024 | 12.60 |
| | | | 0.5 | 8128.20 | 0.023 | 14.51 |
| | | | 0.8 | 9053.55 | 0.032 | 15.37 |
| | | 0.5 | 0.2 | 7632.82 | 0.019 | 14.93 |
| | | | 0.5 | 9258.77 | 0.005 | 13.02 |
| | | | 0.8 | 9711.68 | 0.025 | 12.39 |
| | | 0.8 | 0.2 | 9270.24 | 0.015 | 15.21 |
| | | | 0.5 | 7568.96 | 0.009 | 16.68 |
| | | | 0.8 | 9586.75 | 0.018 | 17.45 |
| **mean** | | | | **8742.31** | **0.195** | **14.68** |
| 3 | 10 | 0.2 | 0.2 | 9388.29 | 0.031 | 16.60 |
| | | | 0.5 | 9043.73 | 0.027 | 20.39 |
| | | | 0.8 | 10157.95 | 0.002 | 18.31 |
| | | 0.5 | 0.2 | 10385.71 | 0.008 | 16.51 |
| | | | 0.5 | 10725.89 | 0.002 | 20.41 |
| | | | 0.8 | 9349.84 | 0.002 | 19.53 |
| | | 0.8 | 0.2 | 10198.34 | 0.002 | 20.41 |
| | | | 0.5 | 9913.64 | 0.002 | 21.35 |
| | | | 0.8 | 9754.75 | 0.002 | 14.63 |
| **mean** | | | | **9879.79** | **0.009** | **18.68** |
| 5 | 10 | 0.2 | 0.2 | 9074.26 | 0.001 | 32.33 |
| | | | 0.5 | 9866.64 | 0.011 | 32.66 |
| | | | 0.8 | 10003.40 | 0.003 | 33.30 |
| | | 0.5 | 0.2 | 10588.96 | 0.008 | 17.57 |
| | | | 0.5 | 10625.64 | 0.003 | 21.96 |
| | | | 0.8 | 10959.66 | 0.012 | 22.81 |
| | | 0.8 | 0.2 | 11533.50 | 0.009 | 29.35 |
| | | | 0.5 | 11764.50 | 0.002 | 31.47 |
| | | | 0.8 | 11405.16 | 0.002 | 30.67 |
| **mean** | | | | **10646.86** | **0.006** | **28.01** |
| **Mean of all** | | | | **9756.32** | **0.011** | **20.46** |

Where MCPU Time is mean of CPU Time for all cases and is calculated to second. Also, $PRE_{avg}$ is mean of all calculated *PREs* for each combination of generated instances. Table 4 shows that the proposed ISETP heuristic is to a large extent a reliable technique which could solve such complicated problems in a very intangible computational time. The latter point, i.e., solving problems in a very small computational time is promising which shows the potentially applicability of the proposed method in solving medium-to-large size instances.

## References


Bank, J. and Werner, F. (2001), "Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties", *Mathematical and Computer Modeling,* Vol. 33 Nos. 4–5, pp. 363-83.

Cheng, T.C.E. and Sin, C.C.S. (1990), "A state-of-the-art review of parallel-machine scheduling research", *European Journal of Operational Research*, Vol. 47, pp. 271–92.

Eswaramurthy, V. P. (2008), "New algorithm for analyzing performance of neighborhood strategies in solving job shop scheduling problems", *Journal of Scientific & Industrial Research*, Vol. 67, pp. 579 –88.

Gurel, S. and Akturk, M. S. (2007), "Optimal allocation and processing time decisions on non-identical parallel CNC machines", *European Journal of Operational Research*, Vol. 183, pp. 591–607.

Hall, N.G. and Posner, M.E. (2001), "Generating experimental date for computational testing with machine scheduling applications", *Operations Research*, Vol. 49, pp. 854–65.

Janiak, A., Janiak, W., Kovalyov, M.Y., Kozan, E. and Pesch, E. (2013), "Parallel machine scheduling and common due window assignment with job independent earliness and tardiness costs". *Information Science*, Vol. 224, No. 1, pp. 109-117.

Kayvanfar, V., Komaki, GH.M. , Aalaei, A., and Zandieh, M. (2014), "Minimizing total tardiness and earliness on unrelated parallel machines with controllable processing times". *Computers & Operations Research*, Vol. 41, pp. 31-43.

Liaw, C.F., Lin, Y.K., Cheng, C.Y. and Chen, M. (2003), "Scheduling unrelated parallel machines to minimize total weighted tardiness", *Computers and Operations Research*, Vol. 30, pp. 1777–89.

Lin, Y. K., Pfund, M.E., and Fowler, J.W. (2011). "Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems". *Computers and Operations Research*, Vol. 38, No. 6, pp. 901-916.

M'Hallah, R., and Al-Khamis, T. (2012). "Minimising total weighted earliness and tardiness on parallel machines using a hybrid heuristic". *International Journal of Production Research*, Vol. 50, No. 10, pp. 2639-2664.

Morton, T. E. & Pentico, D. W. (1993), "*Heuristic Scheduling Systems*", John Wiley & Sons, NY.

Rocha, P. L., Ravetti, M. G., Mateus, G. R. and Pardalos, P. M. (2008), "Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times". *Computers & Operations Research,* Vol. 35 No. 4, pp. 1250-64.

Toksari, M. D. and Güner, E. (2009), "Parallel machine earliness/tardiness scheduling problem under the effects of position based learning and linear/nonlinear deterioration". *Computers & Operations Research*, Vol. 36 No. 8, pp. 2394-417.

Vallada, E., and Ruiz, R. (2011). "A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times". *European Journal of Operations Research*, Vol. 211, No. 3, pp. 612−622.

Ventura, J. A. and Kim, D. (2003), "Parallel machine scheduling with earliness-tardiness penalties and additional resource constraints", *Computers & Operations Research*, Vol. 30 No.13, pp. 1945-58.

Wang, J. B. (2006), "Single machine scheduling with common due date and controllable processing times", *Applied Mathematics and Computation*, Vol. 174, pp. 1245–54.

## Biography

**Vahid Kayvanfar** is a Ph.D. Candidate of Industrial Engineering at Amirkabir University of Technology (Tehran Polytechnic). He got his Master of Science degree in Industrial Engineering from Mazandaran University of Science and Technology in 2011. His research interests mostly include production planning and scheduling in different areas, soft computing, applied operations research and supply chain management. He has published several journal and conference papers in the above-mentioned areas. He has also performed several practical projects in different organizations and taught courses such as operations research, probability & engineering statistics, etc.

**Amin Aalaei** was born at Boroujen district of Chaharmahal & Bakhtiari, Iran, in 1984. He received his B.Sc. in Industrial Engineering from University of Kurdistan in 2007 and Master of Science in Industrial Engineering from Mazandaran University of Science and Technology in 2010. He began his Ph.D. in 2011 at Amirkabir University of Technology (Tehran Polytechnic). His research interests include Cellular Manufacturing Systems, Data Envelopment Analyses, Scheduling, Decision Making Theory and Mathematical Modeling. He has also published more than 15 research papers in reputed international journals.

**Mahtab Hosseininia** is Master of Science in Industrial Engineering from Amirkabir University of Technology (Tehran Polytechnic). She obtained her B.Sc. degree in Industrial Engineering (planning and systems analysis) from Alzahra University. Her research interest follows on the area of operation research, stochastic optimization, game theory, graph theory, facility location, logistics and supply chain management and also operation management. She has worked on the design of distribution centers, physical network configuration and demand planning of oil products throughout supply chain at National Iranian Oil Products Distribution Company (NIOPDC) since 2005.

**Mahdi Rajabi** is Master of Science student in Industrial Engineering at Amirkabir University of Technology (Tehran Polytechnic). He got his B.Sc from Ershad Higher Education Institute in 2012. His research interests involve production planning and scheduling, project management, logistic engineering and supply chain management. He has also done research projects in quality control and risk management area in oilfield at National Iranian Oil Products Distribution Company (NIOPDC).