



## Novel Heuristic Approach for Solving Multi-objective Scheduling Problems

Begard A. Amin

Ayad M. Ramadan

[begardamin1976@gmail.com](mailto:begardamin1976@gmail.com)

[ayad.ramadan@univsul.edu.iq](mailto:ayad.ramadan@univsul.edu.iq)

Mathematics Department, College of Science, Sulaimani University

**Article history: Received 20 December, 2020, Accepted 16, February, 2021, Published in July 2021.**

**Doi: 10.30526/34.3.2677**

### Abstract

In this paper, we studied the scheduling of  $n$  jobs on a single machine. Each of  $n$  jobs is to be processed without interruption and becomes available for processing at time zero. The objective is to find a processing order of the jobs, minimizing the sum of maximum earliness and maximum tardiness. This problem is to minimize the earliness and tardiness values, so this model is equivalent to the just-in-time production system. Our lower bound depended on the decomposition of the problem into two subprograms. We presented a novel heuristic approach to find a near-optimal solution for the problem. This approach depends on finding efficient solutions for two problems. The first problem is minimizing total completion time and maximum tardiness. The second is minimizing total completion time and maximum earliness. We used these efficient solutions to find a near-optimal solution for another problem which is a sum of maximum earliness and maximum tardiness. This means we eliminate the total completion time from the two problems. The algorithm was tested on a set of problems of different  $n$ . Computational results demonstrate the efficiency of the proposed method.

**Keywords:** Scheduling problems, Multi-Objective function, Maximum Tardiness, Maximum Earliness, Heuristic method.

### 1. Introduction

Since 1954, scheduling problems have received much attention in the literature. At first, the researchers considered only one objective function. In practical cases, the decision-maker is obligated to choose only one objective from some objectives [1]. Nowadays, research on multi-criteria scheduling problems has increased. Nagar et al. [2] introduced a survey of multiple and bicriteria problems in scheduling. In general, we have two structures to deal



with conflicting criteria: the hierarchical minimization and the simultaneous minimization [3]. The first one is the primary criterion, and the other is the secondary criterion. In this case, one minimizes the primary criterion and chooses a schedule with a minimum secondary criterion value. In the second approach, the efficient solutions (Pareto set) will be generated, and the decision-maker the one with the best composite objective function [4]. The first paper on a problem of this type was presented by Smith [5]. In this work, the problem of scheduling  $n$  jobs on one machine can be processed at most one job at a time and without interruption. Each job becomes available for processing at time zero, which requires a positive processing time. The objective function is to minimize the sum of maximum earliness and maximum tardiness  $ET_{\max}$ . This problem was first introduced by Amin-Nayeri and Moslehi [6].

We shall use a novel heuristic method to find the best solution for the problem and drive a lower bound which depends on a decomposition of the problem into two subprograms. We present a novel heuristic method to find a best solution depending on a new idea that is inserting an objective function to our problem, and then find efficient solutions for the two problems.

## 2. Notations and Definitions

In this section, we give some notations and definitions

$N$  = the set  $\{1, 2, 3, \dots, n\}$ .

$\Pi$  = the set of permutation schedules.

$\pi$  = a permutation schedule.

$p_j$  = processing time for job  $j$ .

$d_j$  = due date for job  $j$ .

$C_j$  = completion time for job  $j$ .

$E_i = \text{Max} \{d_i - C_i, 0\}$ ; the earliness of job  $i$ .

$E_{\max} = \text{Max} \{E_i\}$ ; the maximum earliness.

$T_i = \text{Max} \{C_i - d_i, 0\}$ ; the tardiness of job  $i$ .

$T_{\max} = \text{Max} \{T_i\}$ ; the maximum tardiness.

MST: (minimum slack times) jobs are sequenced in non-decreasing order of minimum slack times  $s_j$ , where  $s_j = d_j - p_j$ .

SPT: (shortest processing time) they are in non-decreasing order of  $p_j$ .

EDD- rule: (Early due date) they are sequenced in non-decreasing order of  $d_j$ .

LB: lower bound.

UB: upper bound.

**Definition (1)** [7]: Consider a problem  $P$ , a schedule  $\pi \in \Pi$  (where  $\Pi$  is the set of all schedules) is said to be feasible, if it satisfies the constraints of  $P$ .

**Definition (2)** [8]: A feasible schedule  $\pi^*$  is efficient, with respect to the criteria (and ) if there is no any feasible schedule  $\pi$  such  $f(\pi) \leq f(\pi^*)$  and  $g(\pi) \leq g(\pi^*)$ , and at least one of the inequalities is strict.

**3. Problem Formulation**

In this study, we consider the one machine scheduling problem with multi-objective function which is minimizing the sum of the maximum earliness and tardiness, denoted by  $1 // E_{max} + T_{max}$ , for simplicity it is  $1 // ET_{max}$ .

A set  $N = \{1, 2, \dots, n\}$  of  $n$  independent jobs that has to be scheduled on a single machine to minimize a criterion. Job  $j$  ( $j = 1, 2, \dots, n$ ) is to be processed on a single machine without interruption that handles only one job at a time, requires processing time  $p_j$ , and due date  $d_j$ . For a sequence  $\pi$  of the jobs, earliness  $E_{\pi(j)}$ , and the tardiness  $T_{\pi(j)}$  are given by:

$$E_{\pi(j)} = \max \{d_{\pi(j)} - C_{\pi(j)}, 0\}, \quad j = 1, 2, \dots, n.$$

$$T_{\pi(j)} = \max \{C_{\pi(j)} - d_{\pi(j)}, 0\}, \quad j = 1, 2, \dots, n.$$

So the mathematical form of this problem can be formulated as:

$$Z = \min \{E_{max}(\pi) + T_{max}(\pi)\}$$

s. t.

$E_{\pi(j)} \geq 0,$	$j = 1, 2, \dots, n.$	}	... ( $p_1$ )
$T_{\pi(j)} \geq 0,$	$j = 1, 2, \dots, n.$		
$E_{\pi(j)} \geq d_{\pi(j)} - C_{\pi(j)},$	$j = 1, 2, \dots, n.$		
$T_{\pi(j)} \geq C_{\pi(j)} - d_{\pi(j)},$	$j = 1, 2, \dots, n.$		

To find a sequence  $\pi$  that minimizes  $Z$ , the schedules that minimize  $E_{max}$  and  $T_{max}$  are referred to as EDD and MST rules respectively [9]. For the composition of these two functions, we cannot find optimal solution by any direct rule.

**4. Analysis of the Problem with Special Cases**

This problem has been studied in various conditions and compositions. The problem was studied at first by Amin and Moslehi [6], they studied the problem of sequencing a single machine to find an optimal sequence of jobs, also they showed that the algorithm solved 100% of the examples. Since the problem is NP-hard (No Bounded Polynomial Exists) [10], several heuristic approaches have been developed for the problem. Abdul-Razaq and Akram used local search algorithm to solve a multi-criteria function of five objectives included maximum earliness and maximum tardiness [11]. Abdullah considered maximum earliness and maximum tardiness as primary criteria [12].

The problem is also considered with a common due date. In a known sequence, Tavakkoli-Moghaddam et al. [10] found the best value of an idle insert. Tavakkoli-Moghaddam et al. presented also a branch-and-bound algorithm to solve the problem, with an idle insert [13]. Mahnam and Moslehi presented an efficient branch-and-bound algorithm to minimize the problem on a single machine with unequal release times and no unforced idle time [14]. Moslehi, et al., showed that the problem is irregular; so many properties of regular will be missed for the present objective function. They presented a heuristic method using a novel branch-and-bound algorithm to minimize the sum of maximum earliness and tardiness on a single-machine scheduling problem where the due dates were distinct [15].

### 5. Algorithms for Solving Multi-objective Problems

In this section, we introduce two problems that concerning our problem. They have three criteria, namely  $\sum_{j=1}^n C_j$ ,  $T_{max}$  and  $E_{max}$ .

#### 5.1 1/ / F ( $\sum_{j=1}^n C_j, T_{max}$ ) (Total Completion Time and Maximum Tardiness)

This problem can be written as:

$$\begin{array}{l} \text{Min } \sum_{j=1}^n C_j \\ \text{and} \\ \text{Min } T_{max} \end{array} \left. \vphantom{\begin{array}{l} \text{Min } \sum_{j=1}^n C_j \\ \text{and} \\ \text{Min } T_{max} \end{array}} \right\} \dots (p_2)$$

V. Wassenhove and Gelders [8] found all efficient solutions for  $(p_2)$ .

#### 5.2 1/ / F ( $\sum_{j=1}^n C_j, E_{max}$ ) (Total Completion Time and Maximum Earliness)

This problem can be written as:

$$\begin{array}{l} \text{Min } \sum_{j=1}^n C_j \\ \text{and} \\ \text{Min } E_{max} \end{array} \left. \vphantom{\begin{array}{l} \text{Min } \sum_{j=1}^n C_j \\ \text{and} \\ \text{Min } E_{max} \end{array}} \right\} \dots (p_3)$$

Hoogeveen and Van de Velde [16] presented an algorithm in which they found all the efficient solutions (Pareto set) for  $(p_3)$ .

### 6. A Novel Heuristic Approach

Two methods have been used to solve multi-objective scheduling problem exact and approximation method. The exact solutions are obtained by enumeration method, Branch and bound method, and dynamic programming method. The approximate solutions are obtained by local search methods which form a very general class of heuristic methods [9].

Here we will give a novel heuristic method to find a best solution for the  $(p_1)$  problem. Let  $S$  be a set of all efficient solutions for the problem  $(p_2)$ , and let  $S_1$  be a set of all efficient solutions for the problem  $(p_3)$ .

**Proposition (1):**  $S \cap S_1 \neq \varnothing$ .

**Proof:** Since SPT-rule is one of the efficient solutions for  $(p_2)$ , and also, SPT-rule is one of the efficient solutions for  $(p_4)$ , so SPT-rule is one of the common sequences of  $(S \cap S_1)$ . ■

For each  $\pi \in (S \cup S_1)$ , find  $(E_{max}(\pi)+T_{max}(\pi))$ . This construct a set of solutions, and one of them may be an optimal solution for problem  $(p_1)$ . In general, one cannot guarantee to find finding an optimal solution form the set.

#### 6.1 Special Cases of the Problem $(p_1)$

**Case (1):** If a schedule  $\pi$  gives SPT-rule and EDD-rule at the same, then  $\pi$  is optimal schedule for  $(p_2)$ , and produces one efficient solution.

**Proof:** Directly from the definition of SPT-rule and EDD-rule. ■

**Case (2):** If a schedule  $\pi$  gives SPT-rule and MST-rule at the same, then  $\pi$  is optimal schedule for  $(p_3)$ , and produces one efficient solution.

**Proof:** Directly from the definition of SPT-rule and MST-rule. ■

**Case (3):** If a schedule  $\pi$  gives EDD-rule and MST-rule at the same time, then  $\pi$  is optimal schedule for  $(p_1)$ , and produces the only efficient solution for  $(p_1)$ .

**Proof:** Directly from the definition of EDD-rule and MST-rule. ■

**Proposition (2):** There exists an optimal solution  $\pi$  such that  $\pi \notin (S \cup S_1)$ .

**Proof:** Since in general EDD-rule and MST- rule are not efficient solutions for the problems  $(p_2)$  and  $(p_3)$  respectively.

Let  $\pi$  be the EDD-rule, and let  $\pi^*$  be another schedule with  $T_{\max}(\pi^*) = T_{\max}(\text{EDD})$ . If  $\pi^*$  is an efficient solution for  $(p_2)$ , this means  $\pi^* \in S$ , and dominates  $\pi$ , because  $\sum_{j=1}^n C_j(\pi^*) < \sum_{j=1}^n C_j(\pi)$ . At the same time, it is possible that  $E_{\max}(\pi^*) > E_{\max}(\pi)$ , so  $\pi$  is an efficient solution for  $(p_1)$  (Optimal) and dominates  $\pi^*$ . ■

## 6.2 Lower Bound (LB)

Deriving lower bounds for a multiple objective function is difficult, because we have to minimize the objectives at the same time, so we introduce here a lower bound for our problem as follows

### 6.2.1 Derive a Lower Bound (LB)

Consider again the problem  $(p_1)$ . If we have only one objective function,  $E_{\max}$  problem, or  $T_{\max}$  problem, then we can solve the problems by MST-rule and EDD- rule respectively. This technique is called the decomposition of the problem [13]. So

$$LB = E_{\max}(\text{MST}) + T_{\max}(\text{EDD}).$$

## 6.3 Optimal Solution and Efficient Solutions

It is clear that the optimal solution is greater than or equal to the lower bound. Define the upper bound by  $UB = E_{\max}(\text{MST}) + T_{\max}(\text{MST})$ . The relation between the optimal value (opt.), lower bound (LB) and efficient solutions for  $(p_1)$  is presented below.

**Theorem:** There exists an integer  $r \geq 0$  such that  $LB + r = \text{opt.}$ , and  $r \in [N_1 - 1, N_2 + 1]$ , where,  $N_1 =$  number of efficient solutions,  $N_2 = T_{\max}(\text{MST}) - T_{\max}(\text{EDD})$ .

**Proof:** Since  $LB \leq \text{opt.}$ , so there exists an integer  $r \geq 0$  such that  $LB + r = \text{opt.}$ , so the first part of the theorem is proved. Now, we show that  $r \in [N_1 - 1, N_2 + 1]$  or that  $N_1 - 1 \leq r \leq N_2 + 1$ .

$$\begin{aligned} \text{Now } LB + r &= \text{opt.}, \text{ thus } r = \text{opt.} - LB \leq UB - LB \\ &= E_{\max}(\text{MST}) + T_{\max}(\text{MST}) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}). \\ &= T_{\max}(\text{MST}) - T_{\max}(\text{EDD}) = N_2 \leq N_2 + 1. \text{ Hence, } r \leq N_2 + 1. \end{aligned}$$

We will prove  $N_1 - 1 \leq r$  by mathematical induction on  $N_1$ . Note that  $N_1$  is not known.

**If  $N_1 = 1$ ,** that is, there is only one efficient solution which is EDD (not EDD-rule) as well as MST (not MST-rule), then  $r = \text{opt.} - LB$

$$\begin{aligned} &= E_{\max}(\text{Opt.}) + T_{\max}(\text{Opt.}) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) \\ &= E_{\max}(\text{MST}) + T_{\max}(\text{EDD}) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) = 0. \end{aligned}$$

Thus  $N_1 - 1 \leq r \leq N_2 + 1$ . That is  $r \in [N_1 - 1, N_2 + 1]$ , and so the theorem is true for  $N_1 = 1$ .

If  $N_1 = 2$ , i.e., the number of efficient solutions is two which are  $(\sigma)$  and  $(\sigma_1)$  such that  $E_{\max}(\sigma) = E_{\max}(\text{MST})$  and  $T_{\max}(\sigma_1) = T_{\max}(\text{EDD})$ , since  $N_1 = 2$  implies that  $N_1 - 1 = 1$ .

Now if  $(\sigma)$  is optimal then:

$$\begin{aligned} r &= E_{\max}(\sigma) + T_{\max}(\sigma) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) \\ &= E_{\max}(\text{MST}) + T_{\max}(\sigma) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) \\ &= T_{\max}(\sigma) - T_{\max}(\text{EDD}) \geq 1 = N_1 - 1. \text{ Hence } N_1 - 1 \leq r \leq N_2 + 1. \end{aligned}$$

Now if  $(\sigma_1)$  is optimal then:

$$\begin{aligned} r &= E_{\max}(\sigma_1) + T_{\max}(\sigma_1) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) \\ &= E_{\max}(\sigma_1) + T_{\max}(\text{EDD}) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) \\ &= E_{\max}(\sigma_1) - E_{\max}(\text{MST}) \geq 1 = N_1 - 1. \text{ Hence } N_1 - 1 \leq r \leq N_2 + 1. \end{aligned}$$

So,  $r \in [N_1 - 1, N_2 + 1]$  and the theorem is true for  $N_1 = 2$ .

If  $N_1 = 3$ , i.e., there are three efficient solutions  $(\sigma)$ ,  $(\sigma_1)$  and  $(\sigma_2)$  such that  $E_{\max}(\sigma) = E_{\max}(\text{MST})$  and  $T_{\max}(\sigma_2) = T_{\max}(\text{EDD})$ . Since  $N_1 = 3$  implies that  $N_1 - 1 = 2$ .

Now if  $(\sigma)$  is optimal then:

$$\begin{aligned} r &= E_{\max}(\sigma) + T_{\max}(\sigma) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) \\ &= E_{\max}(\text{MST}) + T_{\max}(\sigma) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) \\ &= T_{\max}(\sigma) - T_{\max}(\text{EDD}) \geq 2 = N_1 - 1. \text{ Hence } N_1 - 1 \leq r \leq N_2 + 1. \end{aligned}$$

Now if  $(\sigma_1)$  is optimal then:

$$\begin{aligned} r &= E_{\max}(\sigma_1) + T_{\max}(\sigma_1) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) \\ &= E_{\max}(\sigma_1) - E_{\max}(\text{MST}) + T_{\max}(\sigma_1) - T_{\max}(\text{EDD}) \\ &\geq 1 + 1 = 2 = N_1 - 1. \text{ Hence } N_1 - 1 \leq r \leq N_2 + 1. \end{aligned}$$

Finally if  $(\sigma_2)$  is optimal, then:

$$\begin{aligned} r &= E_{\max}(\sigma_2) + T_{\max}(\sigma_2) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) = E_{\max}(\sigma_2) - \\ &E_{\max}(\text{MST}) + T_{\max}(\text{EDD}) - T_{\max}(\text{EDD}) \\ &\geq 1 + 1 = 2 = N_1 - 1. \text{ Hence } N_1 - 1 \leq r \leq N_2 + 1. \end{aligned}$$

So  $r \in [N_1 - 1, N_2 + 1]$  and it is true for  $N_1 = 3$ .

Suppose the theorem is true for  $N_1 = k$ , i.e., the theorem is true for the  $k$  efficient solutions  $(\sigma)$ ,  $(\sigma_1)$ , ...,  $(\sigma_{k-1})$ , that is, for these  $k$  efficient solutions  $N_1 - 1 \leq r \leq N_2 + 1$ . Let  $N_1 = k + 1$ , that is, there is  $k + 1$  efficient solutions  $(\sigma)$ ,  $(\sigma_1)$ , ...,  $(\sigma_{k-1})$ ,  $(\sigma_k)$ , if any one of the first  $k$  efficient solutions  $(\sigma)$ ,  $(\sigma_1)$ , ...,  $(\sigma_{k-1})$ , is optimal then since it is true for  $N_1 = k$ , we get  $N_1 - 1 \leq r$ , and hence,  $N_1 - 1 \leq r \leq N_2 + 1$ , and if the last efficient solution  $(\sigma_k)$  is optimal, then:

$$\begin{aligned} r &= E_{\max}(\sigma_k) + T_{\max}(\sigma_k) - E_{\max}(\text{MST}) - T_{\max}(\text{EDD}) \\ &= E_{\max}(\sigma_k) - E_{\max}(\text{MST}) \geq k = k + 1 - 1 = N_1 - 1, \text{ thus } N_1 - 1 \leq r \leq N_2 + 1. \end{aligned}$$

Thus it is true for  $N_1 = k + 1$  which completes the proof. ■

**Corollary:** If  $T_{\max}(\text{MST}) = T_{\max}(\text{EDD})$ , then  $r = 0$ .

**Proof:** Since  $T_{\max}(\text{MST}) = T_{\max}(\text{EDD})$  so, we have a schedule  $\pi$  (say) which gives EDD-rule and MST-rule at the same time, and  $\pi$  is also optimal schedule. The lower bound is defined as:

$$\text{LB} = E_{\max}(\text{MST}) + T_{\max}(\text{EDD}). \text{ Therefore, } \text{LB} = \text{opt. this implies that } r = 0. \blacksquare$$

#### 6.4 Modified Smith's Algorithm

Smith presented an algorithm to solve a multi-criteria problem [5]. We modify it to find an optimal solution for a given  $T_{\max}$ .

**Step (1):** Set  $R = \sum_{j=1}^n p_j$ ,  $N = \{1, 2, \dots, n\}$ ,  $k = n$ .

**Step (2):** Find a job  $j^*$  such that  $p_{j^*} \geq p_j$ ,  $D_{j^*} \geq R$ , and

$R - d_j \leq T_{\max}$  (EDD),  $j$  and  $j^* \in N$ .

(Break ties arbitrarily). Assign job  $j^*$  to position  $k$ .

**Step (3):** Set  $R = R - p_j$ ,  $N = N - \{j^*\}$ ,  $k = k - 1$ . If  $k = 0$ , stop. Else, go to step (2).

This theorem finds the difference between the LB and the opt. solution. We use this theorem to find a heuristic method and then the near optimal solution for  $(p_1)$ .

Let  $S_1 = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$  which means that there are  $k$  efficient solutions for the problem  $(p_3)$ . Also let  $S = \{\sigma^*_1, \sigma^*_2, \dots, \sigma^*_{k_1}\}$  which means that there are  $k_1$  efficient solutions for the problem  $(p_2)$ . In  $S_1$ , there exists  $i$  ( $i = 1, \dots, k$ ) such that  $E_{\max}(\sigma_i) = E_{\max}(\text{MST})$ , and there exists also in  $S$   $j$  ( $j = 1, \dots, k_1$ ) such that  $T_{\max}(\sigma^*_j) = T_{\max}(\text{EDD})$ . Compute  $z_i = E_{\max}(\sigma_i) + T_{\max}(\sigma_i) \forall i, i = 1, \dots, k$ . Also compute:  $z_j = E_{\max}(\sigma^*_j) + T_{\max}(\sigma^*_j) \forall j, j = 1, \dots, k_1$ .

We get a set of solutions  $z_i$  and  $z_j$  which are upper bounds of the problem  $(p_1)$ . This means that they are greater than or equal to the optimal solution. Let  $z^*$  be the minimum one of  $z_i$  and  $z_j$ .

From the theorem, we have  $\text{opt.} = \text{LB} + r$ , where  $r \in [N_1 - 1, N_2 + 1]$ . Now we have the possible cases for the optimal.

If  $N_1 = 1$ , implies that  $r \in [0, N_2 + 1]$ ,  $\text{opt.} = \text{LB} + 0$ .

If  $N_1 = 2$ , implies that  $r \in [1, N_2 + 1]$ ,  $\text{opt.} = \text{LB} + 1$ , and so on. Since we have  $z^*$  as the minimum upper bound, so if  $\text{opt.} = \text{LB} + 1 > z^*$  stop the process.

For the acceptable values of  $r$  we start with  $r = 0$ . It is a special case where  $T_{\max}(\text{MST}) = T_{\max}(\text{EDD})$ , in this case, one can find the optimal solution directly. If not. We check for  $r = 1$ . If there exists a sequence  $\pi^* \in S_1 \cup S$  such that  $E_{\max}(\pi^*) + T_{\max}(\pi^*) = \text{LB} + 1$ , so it is optimal, otherwise solve  $(p_1)$  using modified Smith's algorithm with given  $T_{\max}(\text{EDD})$ . If a solution exists, it is optimal otherwise check for  $r = 2$ .

### The Proposed Algorithm:

**Step (1):** Find efficient solutions for  $(p_3)$ , say  $S_1 = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$  and for  $(p_2)$ , say  $S = \{\sigma^*_1, \sigma^*_2, \dots, \sigma^*_{k_1}\}$ .  $N_1$  is the number of efficient solutions of  $(p_1)$ ,  $N_2 = T_{\max}(\text{MST}) - T_{\max}(\text{EDD})$  and  $\text{LB} = E_{\max}(\text{MST}) + T_{\max}(\text{EDD})$ .

**Step (2):** Compute  $z_i = E_{\max}(\sigma_i) + T_{\max}(\sigma_i) \forall i, i = 1, \dots, k$ , and compute  $z_j = E_{\max}(\sigma^*_j) + T_{\max}(\sigma^*_j) \forall j, j = 1, \dots, k_1$ .  $z^*$  is minimum of  $z_i$  and  $z_j$ .

**Step (3):** If  $T_{\max}(\text{MST}) = T_{\max}(\text{EDD})$ , then EDD-rule as well MST-rule give optimal solution. Go to step (7). Else, go to step (4).

**Step (4):** Find values of  $r$  such that  $r \in [N_1 - 1, N_2 + 1]$ , stop if  $r > N_2 + 1$  or  $r > z^*$ ,  $r = 1$ .

**Step (5):** If there exists a sequence  $\pi^* \in S_1 \cup S$  such that  $E_{\max}(\pi^*) + T_{\max}(\pi^*) = \text{LB} + r$ , so it is optimal and go to step (7). Else, go to step (6).

**Step (6):** Use modified Smith’s algorithm for given  $T_{\max}(\text{EDD})$ . If a solution exists, it is optimal. Go to step (7). Else,  $r = r + 1$ . Go to step (5).

**Step (7):** Stop.

**Example:** Consider a 4 job problem with the following given data

j	1	2	3	4
$p_j$	5	3	2	7
$d_j$	8	6	11	7

At first, we find efficient solutions for the problems  $(p_2)$  and  $(p_3)$  by suitable algorithms. The results are in the following table. We characterize each efficient solution for each problem by  $\bullet$ , and the other solutions are the corresponding solutions.

Numbers	Schedules	$(\sum_{j=1}^4 C_j, E_{\max})$	$(\sum_{j=1}^4 C_j, T_{\max})$	$(E_{\max} + T_{\max})$
1	(3, 2, 1, 4)	$\bullet$ (34, 9)	$\bullet$ (34, 10)	19
2	(2, 3, 1, 4)	$\bullet$ (35, 6)	(35, 10)	16
3	(3, 1, 2, 4)	(36, 9)	$\bullet$ (36, 9)	18
4	(2, 1, 3, 4)	$\bullet$ (38, 3)	(38, 10)	13
5	(2, 1, 4, 3)	(43, 3)	$\bullet$ (43, 8)	11
6	(2, 4, 1, 3)	(45, 3)	$\bullet$ (45, 7)	10
7	(4, 2, 3, 1)	$\bullet$ (46, 0)	(46, 9)	9

$$S_1 = \{(3, 2, 1, 4), (2, 3, 1, 4), (2, 1, 3, 4), (4, 2, 3, 1)\},$$

$$S = \{(3, 2, 1, 4), (3, 1, 2, 4), (2, 1, 4, 3), (2, 4, 1, 3)\}, k = 4 \text{ and } k_1 = 4.$$

In  $S_1$  schedule number 7 gives  $E_{\max}(7) = E_{\max}(\text{MST}) = 0$ , and in  $S$  schedule number 6 gives  $T_{\max}(6) = T_{\max}(\text{EDD}) = 7$  (in this example it is EDD-rule).

Now find the set of solutions  $z_i$  and  $z_j$  which are in column 5.  $z^* = 9$

To apply the theorem  $LB = 7 + 0 = 7, N_2 = 7 - 7 = 0$ , so  $r \in [N_1 - 1, 1]$ .

If  $N_1 = 1$ , implies that  $r \in [0, 1], \text{opt.} = LB + 0 = 7$ .

If  $N_1 = 2$ , implies that  $r \in [1, 1], \text{opt.} = LB + 1 = 8$ .

The possible cases for the optimal value are 7, 8, and 9.

Start with  $r = 0$ . Since there exists no  $\pi^* \in S_1 \cup S$  such that  $E_{\max}(\pi^*) + T_{\max}(\pi^*) = LB + 0 = 7$ , use the modified Smith’s algorithm, where  $T_{\max} = T_{\max}(\text{EDD})$ . The solution is (4, 2, 1, 3) with  $T_{\max} = 7$  and  $E_{\max} = 0$ .

### 7. Computational Results

Here, we will introduce our results via computational tests to show the efficiency of the proposed algorithm. The problems are generated as follows: an integer processing time  $p_j$  generated from the uniform distribution  $[1, 10]$ . Also, an integer due date  $d_j$  is generated from the uniform distribution  $[0, \sum_{j=1}^n p_j]$ .

The proposed algorithm was running and coding it in Matlab 8.1 (R2013a) and implemented on Intel (R) Pentium (R) CPU 2117U@ 1.80 GHZ, with RAM 4.00 GB personal computer.

## 8. Conclusions and Future Works

We presented a novel heuristic algorithm to minimize the sum of maximum earliness and tardiness in single-machine scheduling  $ET_{max}$ . The algorithm depended on a new idea that is presented for the first time. The algorithm was easy to implement and had a simple structure comparing with other used algorithms such as branch and bound algorithm. To evaluate the efficiency of the proposed algorithm, the algorithm was tested on (25) examples for different  $n$ . For  $n = 3$ , it solves 100% of the instances. For  $n = 4$ , it solves 100% of the instances. For  $n = 5$ , it solves also 100% of the instances. Computational results showed the ability of the algorithm.

An important area for future research is to use the presented idea for other multi-objective problems with a variety of environments and conditions.

## References

1. T'kindt, V.; Billaut, J.-C. *Multicriteria Scheduling: Theory, Models and Algorithms*, 2<sup>nd</sup>; Springer: Berlin, 2002; 9783540282303.
2. Joksch, H. C. Constraints objectives, efficient solutions and sub optimization in mathematical programming. *Journal of Institutional and Theoretical Economics*. **1996**, *122*, 5-13.
3. Hoogeveen, J.A. Single-Machine Bicriteria Scheduling. Ph.D. Thesis. CWI, Amsterdam. **1992**.
4. Hoogeveen, J. A. Minimizing maximum promptness and maximum lateness on a single machine. *Mathematics of Operation Research*. **1996**, *21*, 100-114, <https://doi.org/10.1287/moor.2018.0943>.
5. Ayad, M. Ramadan. Range of lower bounds. *Applied Mathematics and Computation*. **2011**, *218*, 1008-1011, <https://doi.org/10.1016/j.amc.2020.125904>
6. Amin-Nayeri, M.R.; Moslehi, G. Optimal algorithm for single machine sequencing to minimize early/tardy cost. *ESTEGHLAL Journal of Engineering*. **2000**, *19*, 35–48.
7. Jouni, L. Multi-objective nonlinear Pareto-optimization. Lappeenranta University of Technology, *Laboratory of Information Processing*. **2000**.
8. Van Wassenhove, L. N.; Gelders, F. Solving a bicriterion scheduling problem. *European Journal of Operational Research*. **1980**, *4*, 42-48, <https://doi.org/10.1016/j.ejor.2021.01.025>.
9. Abdul-Razaq, T. S.; Hussam Abid Ali Mohammed. Exact and approximation approaches for solving multiple objective function. *Al-Mustansiriyah Journal of Science*. **2016**, *27*, 89-97, <http://dx.doi.org/10.23851/mjs.v31i4.897>.
10. Moghaddam, R.T.; Moslehi, G.; Vasei, M.; Azaron, A. Optimal scheduling for a single machine to minimize the sum of maximum earliness and tardiness considering idle insert. *Applied Mathematics and Computation*. **2005**, *167*, 1430-1450, <https://doi.org/10.1016/j.amc.2020.125904>.

11. Abdul-Razaq, T. S.; Akram, A. O. Local search algorithms for multi-criteria single machine scheduling problem. *Ibn Al-Haitham Journal for Pure and Applied Science (Special issue)*. **2017**, 436-451, <http://dx.doi.org/10.30526/2017.IHSCICONF.1817>.
12. Abdullah, H. F. Multicriteria scheduling problems to minimize total tardiness subject to maximum earliness or tardiness. *Ibn Al-Haitham Journal for Pure and Applied Science*. **2010**, 23, 311-320, <http://dx.doi.org/10.30526/2017.IHSCICONF.1817>.
13. Tavakkoli-Moghaddam, R.; Moslehi, G.; Vasei, M.; Azaron, A. a branch-and-bound algorithm for a single machine sequencing to minimize the sum of maximum earliness and tardiness with idle insert. *Applied Mathematics and Computation*. **2006**, 174, 388–408, <https://doi.org/10.1016/j.amc.2020.125904>.
14. Mahnam, M.; Moslehi, G. A branch and bound algorithm for minimizing the sum of maximum earliness and tardiness with unequal release times. *Engineering Optimization*. **2009**, 41, 521–536.
15. Moslehi, G.; Mahnam, M.; Amin-Nayeri, M.; Azaron, A. A branch-and-bound algorithm to minimize the sum of maximum earliness and tardiness in the single machine. *Int. Journal Operational Research*. **2010**, 8, 458-482.
16. Hoogeveen, J.A.; Van de Velde, S.L. Minimizing total completion time and maximum cost simultaneously is solvable in polynomial time. *Operations Research Letters*. **1995**, 17, 205– 208.