ISSN 2303-4521
Original Research

Periodicals of Engineering and Natural Sciences
Vol. 11, No. 3, May 2023, pp.46-57

Solving the multi-criteria: total completion time, total late work, and maximum earliness problem

Nagham Muosa Neamah[1,2*], Bayda Atiya Kalaf[2]
[1]Mathematics Department, College of Science for Women University of Baghdad, Baghdad, Iraq
[2]Mathematics Department, College of Education for Pure Science Ibn-Al-Haitham, University of Baghdad, Baghdad, Iraq

ABSTRACT

Within this research, The problem of scheduling jobs on a single machine is the subject of study to minimize the multi-criteria and multi-objective functions. The first problem, minimizing the multi-criteria, which include Total Completion Time, Total Late Work, and Maximum Earliness Time $(\sum C_j, \sum V_j, E_{max})$, and the second problem, minimizing the multi-objective functions $\sum C_j + \sum V_j + E_{max}$ are the problems at hand in this paper. In this study, a mathematical model is created to address the research problems, and some rules provide efficient (optimal) solutions to these problems. It has also been proven that each optimal solution for $\sum C_j + \sum V_j + E_{max}$ is an efficient solution to the problem $(\sum C_j, \sum V_j, E_{max})$. Because these problems are NP-hard problems so it is difficult to determine the efficient (optimal) solution set for these problems so some special cases are shown and proven which find some efficient (optimal) solutions suitable for the discussed problem, and highlight the significance of the Dominance Rule (DR), which can be applied to this problem to enhance efficient solutions.

Keywords: Maximum Earliness, Multi-Criteria (MC), Multi-Objective (MO), Total Completion Times, Total Late Work.

Corresponding Author:

Nagham Muosa Neamah
Mathematics Department, College of Science for Women
University of Baghdad, Iraq
Baghdad, Iraq
Email: naghamm_math@csw.uobaghdad.edu.iq

1. Introduction

The scheduling problem, an important subject (topic) for operations research (OR), is the focus of this paper. It has the following definition: there are a specified number of jobs, n, each involving one or more processes, and must be scheduled on one or more machines during a predetermined period to minimize the specified objective function [1]. It has also been defined as, it is putting things (lectures, vehicles, people, tasks, jobs, etc.) into a pattern in time and space so that conditions are met and specific goals are achieved [2]. It also represents the problem of allocating specific functions to a group of machines at the right time under certain constraints [3].

Up until the late 1980s, mainstream research concentrated on a certain single objective problem. When more than one objective (criteria) is needed, scheduling difficulties become more and more complicated in model and solving. It is frequently implausible that different objectives will be best served by the same set of decision variables [4]. As a result, a trade-off exists between the multi-objectives. Multi-objective scheduling problems are the term used to describe this type of problem (referred to as multi-objective scheduling problems) [3]. A set of Pareto optimal solutions (Efficient solutions), rather than a single optimal solution, are established using multi-criteria optimization based on competing objective functions. This collection includes one (or more) solutions that no other better solution (another solution is preferable) with respect to objective functions.

The most important literature survey for the last ten years is that of Doha in 2012[5], which discussed the multi-criteria scheduling problems which are studied on a single machine to come up with a set of efficient solutions to the general problems $1// F(\sum C_j, \sum V_j)$, $1// F(\sum C_j, \sum U_j)$ and others. Ibrahim in (2022) [6]studied the multi-objective problem, which is the, $1// \sum_{=1}^{n}(E_j + T_j + C_j + U_j + V_j)$, $1// \sum_{=1}^{n}(\alpha_j E_j + \beta_j T_j + \theta_j C_j + \gamma_j U_j + \omega_j V_j)$, $1/ S_f/ \sum_{=1}^{n}(\alpha_{jf} E_{jf} + \beta_{jf} T_{jf} + \theta_{jf} C_{jf} + \gamma_{jf} U_{jf} + \omega_{jf} V_{jf})$, also they suggested an Upper Bound (UB) and a Lower Boundary (LB) to be used in Branch And Bound (BAB) method. Ahmed in 2022 [7] studied the multi-criteria$(\sum \boldsymbol{C_j}, \boldsymbol{T_{max}}, \boldsymbol{R_L})$and multi-objective function $(\sum \boldsymbol{C_j} + \boldsymbol{T_{max}} + \boldsymbol{R_L})$and found the optimal solution by using BAB method with and without DR then use some heuristic methods. Hassan et al in 2022 [8] introduced a heuristic algorithm to reduce the $(\sum \boldsymbol{C_j} + \boldsymbol{E_{max}} + \boldsymbol{T_{max}})$ in just one machine scheduling.

This paper displays the tri-criteria scheduling problems and begins with some basic scheduling concepts of multi-criteria problems, and basic rules are given in section (1). Section (2), establishes a mathematical model for problem $1//F(\sum C_j, \sum V_j, E_{max})$ and sub-problem $1//\sum C_j + \sum V_j + E_{max}$, explains the relationship between them, the Dominance Rule is described and proves several rules, and proves special cases for the problems that lead to the efficient and optimal solution to these problems in this section. In section (3) the significant results obtained in the previous section are presented and discussed. The conclusions and lists of future works are given in section (4).

1.1. Some important notations

In this paper, the following notations are used:

$\boldsymbol{N}$: The jobs set s.t. $N = \{1,2, \dots, n\}$.

$\boldsymbol{n}$: Number of available jobs.

$\boldsymbol{p_j}$: The time of the job $j's$ processing.

$\boldsymbol{d_j}$: The Job's due date for j (or the job's due date), the date for finishing the jobs*;* job termination after the deadline is allowed but will result in a penalty.

$\boldsymbol{s_j}$ **:** The Job's slack time for j s.t. $s_j = d_j - p_j$.

$\boldsymbol{C_j}$: The job's completion time for j, where $C_j = \sum_{k=1}^{j} p_k$.

$\boldsymbol{L_j}$ **:** The lateness time of jobs, s.t. $L_j = -(d_j - C_j) = C_j - d_j$.

$\boldsymbol{E_j}$: The job's earliness time for j , s.t. $E_j = max\,\{0, -L_j\}$.

$\boldsymbol{T_j}$: The job j's tardiness, s.t. $T_j = max\,\{0, L_j\}$.

$\boldsymbol{V_j}$: A late work of a job j, s.t. $V_j = min\,\{T_j, p_j\} = min\{C_j - d_j, p_j\}$.

$\sum \boldsymbol{C_j}$: Total completion time.

$\sum \boldsymbol{V_j}$: Total Late work.

$\boldsymbol{E_{max}}$: Maximum earliness where $E_{max} = max_{j \in N}\{E_j\}$.

Shortest processing time (SPT): Jobs are Sequencing in non-decreasing order of the processing times p_j (i.e. $p_1 \le p_2 \le \dots \le p_n$), this rule is well known to solve $\sum C_j$ for problem $1// \sum C_j$ (Smith 1956)[9].

Earliest due date (EDD): Jobs are sequenced in non-decreasing order of their due dates d_j(i.e. $d_1 \le d_2 \le \dots \le d_n$), this rule is used to solve the problem $1// T_{max}$ [10].

Minimum Slack Time (MST): Jobs are sequenced in non-decreasing order of their slack time $s_j = d_j - p_j$ (i.e. $s_1 \le s_2 \le \dots \le s_n$). To minimize E_{max} using this rule (Hoogeveen 1990) [11] .

EFSO: Efficient solution. **EFSQ**: Efficient sequence.

OPSO: Optimal solution. **OPSE**: Optimal sequence.

1.2. Machine Scheduling Problem (MSP)

This paper needs some basic definitions and theories: [12],[13],[14]

Definition (1.2.1) Objective Function[12]**:** The objective function of an **MSP** is one that may be either minimized or maximized under all possible constraints.

Definition (1.2.2) Feasible Schedule (FS)[13]**:** A schedule is considered feasible if it satisfies several conditions associated with a particular type of problem and two general constraints.

Definition (1.2.3) Efficient Solution(EFSO)[8]: A schedule α^* is a feasible schedule (FS) known as "Pareto optimal" or "non-dominated" If there is absolutely no a feasible schedule (FS) α, then the list of schedules that are feasible with relation to the criteria $f, g,$ and h such that $f(\alpha) \le f(\alpha^*)$, $g(\alpha) \le g(\alpha^*)$ and $h(\alpha) \le h(\alpha^*)$, are satisfied for at least one of the inequalities.

Definition (1.2.4)[14][15]: The solution, which is impossible (that cannot be) to create and improve without worsening the other objectives is referred to as an " optimize" in a multi-criteria decision–making problem.

Definition (1.2.5)[14]: The $\boldsymbol{\sigma}^*$ the schedule is considered to be **optimal** if there is no other schedule σ satisfying $f_j(\sigma) \le f_j(\sigma^*), j = 1, \dots, k$ (k: number of criteria), assuming strict inequality for at least one of the aforementioned conditions. If not, then σ is considered to be dominant over σ^*.

Definition(1.2.6) [16]**:** The graph G represents a finite number of nodes or vertices V and a finite number of edge, connecting two vertices, and the edge connecting the vertex to itself is called the loop .

Definition(1.2.7) [16]**:** If n vertices make up a graph called G, then $A(G) = [a_{ij}]$ be the matrix (which is called adjacency matrix), whose $i^{th}$and j^{th} element is 1 if there is at least one edge between two vertices v_1 and v_2 and zero otherwise , $a_{ij} = \begin{cases} 0, \text{if } i = j \text{ or } i \nrightarrow j \\ 1, \text{if } i \rightarrow j \\ a_{ij}, \text{ otherowise} \end{cases}$.

2. Method

This section is dedicated to studying the mathematical model proposed to address the research problem $1//F(\sum C_j, \sum V_j, E_{max})$ and sub-problem $1// \sum C_j + \sum V_j + E_{max}$. In addition, determine the relationship between the problem $1//F(\sum C_j, \sum V_j, E_{max})$ and the sub-problem $1// \sum C_j + \sum V_j + E_{max}$. Dominance Rules have been used to indicate whether a specific node can be neglected to determine an efficient solution to the proposed problem. Also in this part, some special cases of problems $1// \sum C_j + \sum V_j + E_{max}$ and $1// \sum C_j + \sum V_j + E_{max}$ are proved, leading to efficient and optimal solutions to these problems, respectively.

2.1. Mathematical formulation of the suggested problem

In this section, the three-criteria scheduling problem to be studied will be described. Let the number of jobs available at time 0 be represented by $N = \{1,2, \dots, n\}$, (i.e, $r_j = 0\ \forall j \in N$) and need processing on just one machine. There is a due date d_j and a processing time p_j for every job j, given a sequence of jobs $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$, generate the earliest completion time $C_j = \sum_{k=1}^{j} p_{\sigma_k}$, the $T_j = max\{C_j - d_{\sigma_j}, 0\}$ job j's tardiness, the earliness of job j, $E_j = max\{d_{\sigma_j} - C_j, 0\}$, and $V_j = min\{T_j, p_{\sigma_j}\}$ the job j's late work.

The aim of this problem is finding a schedule σ ∈ $\mathcal{S}$ (where $\mathcal{S}$ is the set of all possible feasible schedules) that minimizes the tri-criteria $(\sum C_j, \sum V_j, E_{max})$, which is denoted by $(TCTVE)$, can be mathematically formulated as shown below:

$F_{CVE} = Min(\sum C_j, \sum V_j, E_{max})$.

Subject to

$$\left.\begin{array}{ll} C_j \geq p_{\sigma_j}, & j = 1, \ldots, n \\ C_j = \sum_{k=1}^{j-1} p_{\sigma k} + p_{\sigma j}, & j = 1,2, \ldots, n \\ T_j \geq C_j - d_{\sigma_j}, & j = 1, \ldots, n \\ E_j \geq d_{\sigma_j} - C_j, & j = 1, \ldots, n \\ V_j = min\left\{T_j, p_{\sigma_j}\right\}, & j = 1,2, .., n \\ V_j \geq 0,\ E_j \geq 0, \text{and } T_j \geq 0, & j = 1, \ldots, n \end{array}\right\} \quad (TCTVE).$$

The σ_j indicates where job j falls in the ordering σ and $\mathcal{S}$ represents the collection of all schedules. Finding all efficient solutions to solve the problem $(TCTVE)$ is challenging, since it's an NP-hard problem (because the problem $1//\sum_{j=1}^{n} V_j$ is NP-hard [1]).

Proposition (1): There is an efficient sequence for the problem $1//F(\sum C_j, \sum V_j, E_{max})$, which satisfies the SPT rule.

Proof: (a) first, assume that $p_i \neq p_j$ for all i, j. The unique sequence SPT, (SPT^*) provides the bare minimum of $\sum C_j$. As a result, no sequence exists $\delta \neq SPT^*$ s.t.

$$\sum C_j(\delta) \leq \sum C_j\,(SPT^*), \sum V_j\,(\delta) \leq \sum V_j\,(SPT^*), \text{and } E_{max}(\delta) \leq E_{max}(SPT^*) \qquad (1)$$

The presence of at least one of the strict inequalities.

(b) If more than one SPT sequence exists in some (jobs with equal processing times), let SPT^* be a sequence satisfying the SPT rule and such that jobs with equal processing times are in EDD (where EDD and MST sequences are identical). If a set of jobs that are to be early or partially early is specified, then this EDD order minimized $\sum V_j$.

Note that if the event is several jobs at the same processing times, the due date is considered identical, or slack times, then SPT^* is not unique. Show that each SPT^* sequencing is an efficient, sequencing that does not satisfy the SPT rule which cannot dominate an SPT^* sequencing by (1). If δ is an SPT sequence, it is not SPT* sequencing, because it cannot dominate SPT^* because

$$\sum C_j(\delta) = \sum C_j(SPT^*), \sum V_j\,(SPT^*) \leq \sum V_j\,(\delta) \text{ and } E_{max}(SPT^*) \leq E_{max}(\delta) \qquad (2)$$

as a result of the EDD rule. Hence each one of the SPT^* sequences are efficient.

The preceding proposition (1) shows that the SPT rule is efficient for problem $1//F(\sum C_j, \sum V_j, E_{max})$ while the EDD rule does not give an efficient solution for the problem $1//F(\sum C_j, \sum V_j, E_{max})$ as demonstrated in the following example:

Example (1): Let's have the following MSP data with $n = 5$:

	Job1	***Job2***	***Job3***	***Job4***	***Job5***
$\boldsymbol{p_j}$	*2*	*5*	*7*	*5*	*8*
$\boldsymbol{d_j}$	*6*	*9*	*8*	*11*	*14*
$\boldsymbol{s_j}$	*4*	*4*	*1*	*6*	*6*

A feasible schedule is provided by the SPT rule(1,2,4,3,5) and (1,4,2,3,5), hence $\left(\sum C_j, \sum V_j, E_{max}\right) =$ (67,16,4) from SPT^* order (1,2,4,3,5) and $\left(\sum C_j, \sum V_j, E_{max}\right) =$ (67,18,4)from SPT order (1,4,2,3,5), it is evident that in (it is obvious that in) SPT^* order the jobs (2,4)with equal processing time are ordered in MST or EDD rule. But EDD rule (1,3,2,4,5) with $\left(\sum C_j, \sum V_j, E_{max}\right) =$ (71,19,4) and MST rule (3,1,2,4,5) with $\left(\sum C_j, \sum V_j, E_{max}\right) =$ (76,20,1) hence SPT^* the order provides an efficient solution to the problem ($TCTVE$).

2.1.1. Sub-problem of ($TCTVE$)

The problem $1//F\left(\sum C_j, \sum V_j, E_{max}\right)$ can deduce a sub-problem (SP), that it minimizes $1//\left(\sum \boldsymbol{C_j} + \sum \boldsymbol{V_j} + \boldsymbol{E_{max}}\right)$.This problem is described as follows:

Assume that σ is any schedule that can be expressed as follows for a certain schedule $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$:

$$\left.\begin{array}{ll} F_{SP} = Min\left(\sum C_j + \sum V_j + E_{max}\right) & \\ \text{subject to} & \\ C_j = \sum_{k=1}^{j} p_{\sigma_k} & j = 1,2,\ldots,n \\ C_j = C_{(j-1)} + p_{\sigma_j} & j = 2,3,\ldots,n \\ E_j \geq d_{\sigma_j} - C_j & j = 1,2,\ldots,n \\ T_j \geq C_j - d_{\sigma_j} & j = 1,2,\ldots,n \\ V_j = min\left\{T_j, p_{\sigma_j}\right\} & j = 1,2,\ldots,n \\ V_j \geq 0,\ E_j \geq 0,\ T_j \geq 0 & j = 1,2,\ldots,n \end{array}\right\} \quad (SP)$$

The objective of the NP-hard problem (SP) is to determine the order of jobs that need to be processed on a single machine in order to minimize the sum of total completion time, total late work, and the maximum earliness jobs.

2.2. Dominance rules (DRs) for MSP

Dominance Rules (DRs) are used efficiently in reducing the current sequences. DRs are used usually to indicate whether a certain node in a BAB method can be eliminated before calculating its (LB). These rules are been useful when a node has an (LB) less than the optimum solution and can be eliminated. When the nodes are dominated by others in the BAB procedure, DRs can be used also to cut these nodes. Such developments may heavily reduce the number of nodes in searching for an efficient solution. Where the DRs are also applicable to such problems [1].

Theorem (1): If $p_i \leq p_k$ and $d_i \leq d_k$ then there is an optimal schedule for the problem (SP) in which the job i processing before job k .

Proof: Suppose there is a sequence $\vartheta = \vartheta_1 i\, k\, \vartheta_2$ and let $\acute{\vartheta} = \vartheta_1\, ki\, \vartheta_2$ be a sequence obtained by interchanging the position of jobs i and k. There are two cases for the sequence ϑ and $\acute{\vartheta}$:

First case: If $p_i \leq p_k$ and $d_i \leq d_k$ implies $s_i \leq s_k$ for every $i, k = 1,2, \ldots, n$, from $p_i \leq p_k$ there are $: \sum_k C_k(\vartheta) \leq \sum_k C_k(\acute{\vartheta})$

From the condition of slack time $s_i \leq s_k$, there are $E_{max}(\vartheta) \leq E_{max}(\acute{\vartheta})$. From $p_i \leq p_k$ and $d_i \leq d_k$, this means $\sum_k V_k(\vartheta) \leq \sum_k V_k(\acute{\vartheta})$. Hence, we have:

$\left(\sum_k C_k(\vartheta) + \sum_k V_k(\vartheta) + E_{max}(\vartheta)\right) \leq \left(\sum_k C_k(\acute{\vartheta}) + \sum_k V_k(\acute{\vartheta}) + E_{max}(\acute{\vartheta})\right).$

Second case: If $p_i \leq p_k$ and $d_i \leq d_k$ implies $s_i > s_k$ for every $i, k = 1,2, \ldots, n$. From $p_i \leq p_k$ we have:

$$\sum_k C_k(\vartheta) \leq \sum_k C_k(\acute{\vartheta}) \quad (3)$$

Equation (3) is satisfied by the condition on processing times, and the addition in cost which is obtained from (3) is equal to $p_k - p_i$, this gives:

$$\sum_k C_k(\vartheta) + P_k - P_i = \sum_k C_k(\acute{\vartheta}) \quad (4)$$

The slack time's condition $s_i > s_k$ implies $E_{max}(\vartheta) > E_{max}(\acute{\vartheta})$. Also, the additional cost $s_i - s_k$ gives:

$E_{max}\left(\acute{\vartheta}\right) + s_i - s_k = E_{max}\left(\vartheta\right)$ (5)
$s_i - s_k = (d_i - p_i) - (d_k - p_k) = (d_i - d_k) + (p_k - p_i) \le p_k - p_i$ (6)
Adding $E_{max}\left(\acute{\vartheta}\right)$ to both sides of (6) we have:
$E_{max}\left(\acute{\vartheta}\right) + s_i - s_k \le E_{max}\left(\acute{\vartheta}\right) + p_k - p_i$, and from (5) we have
$E_{max}\left(\vartheta\right) \le E_{max}\left(\acute{\vartheta}\right) + p_k - p_i$ (7)
Adding $\sum_k C_k(\vartheta)$to both sides of (7) and by (4) we have
$\sum_k C_k\left(\vartheta\right) + E_k(\vartheta) \le \sum_k C_k\left(\acute{\vartheta}\right) + E_{max}\left(\acute{\vartheta}\right)$ (8)
From the conditions $p_i \le p_k$ and $d_i \le d_k$ we have $\sum_k V_k\left(\vartheta\right) \le \sum_k V_k\left(\acute{\vartheta}\right)$. By adding this result to relation (8): $\left(\sum_k C_k(\vartheta) + \sum_k V_k\left(\vartheta\right) + E_{max}\left(\vartheta\right)\right) \le \left(\sum_k C_k(\acute{\vartheta}) + \sum_k V_k\left(\acute{\vartheta}\right) + E_{max}\left(\acute{\vartheta}\right)\right)$. Hence ϑ is better than the sequence $\acute{\vartheta}$ in the two cases and a job i proceed job k in the optimal solution.

Example (2): Let's use MSP with 6 jobs and the following processing time and due date:

	job_1	job_2	job_3	job_4	job_5	job_6
p_j	1	8	10	4	10	9
d_j	14	28	27	23	12	28
s_j	13	20	17	19	3	19

The DRs by using theorem (1) is illustrated in Figure 1.

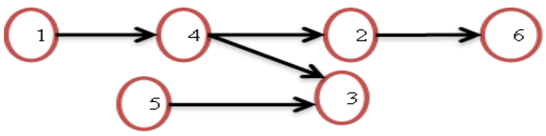

Figure 1. DR is shown in example (2).

Notice that there are (9) DRs: 1→2, 1→ 3 ,1→4, 1→6, 2→ 6, 4→2, 4→ 3, 4→ 6,5→3. with (6) potential sequences some (or all) are governed by the aforementioned DRs listed in Table 1. The adjacency matrix A is as followings:

$$A(G) = \begin{bmatrix} 0 & 1 & 1 & 1 & a_{15} & 1 \\ 0 & 0 & a_{23} & 0 & a_{25} & 1 \\ 0 & a_{32} & 0 & 0 & 0 & a_{36} \\ 0 & 1 & 1 & 0 & a_{45} & 1 \\ a_{51} & a_{52} & 1 & a_{54} & 0 & a_{56} \\ 0 & 0 & a_{63} & 0 & a_{65} & 0 \end{bmatrix}, \text{ where } a_{ji} = \begin{cases} 1, \text{if } a_{ij} = 0 \\ 0, \text{if } a_{ij} = 1 \end{cases}.$$

Table I. The potential efficient sequences are subject to DR in example (2).

	EF.SE. W. DR						$(TCTVE)$	(SP)
Seq	1	2	3	4	5	6	$\left(\sum C_j, \sum V_j, E_{max}\right)$	$\sum C_j + \sum V_j + E_{max}$
1	5	1	4	3	2	6	(136,14,8)	158
2	5	1	4	2	3	6	(134,15,8)	157
3	1	5	4	3	2	6	(127,14,13)	154
4	1	5	4	2	3	6	(125,15,13)	153

5	1	4	5	3	2	6	(121,17,18)	156
6	1	4	5	2	3	6	(119,18,18)	155

Where *EF.SE. W. DR*: Efficient Sequences with DR.

The sequences (1- 6) provide the problem $(TCTVE)$ an efficient value, as can be shown in table (I), observe that sequence number (4) in table (I) provides an optimal value for the problem (SP).

Proposition (2): Each optimal solution for (SP) is an efficient solution to the problem $(TCTVE)$.

Proof: let β be an optimal schedule for (SP). Suppose that β gives no efficient solution for the problem $(TCTVE)$, then there is an efficient schedule say π for $(TCTVE)$-problem such that:

$\sum C_j(\pi) \leq \sum C_j(\beta)$ and $\sum V_j(\pi) \leq \sum V_j(\beta)$ and $E_{max}(\pi) \leq E_{max}(\beta)$.

At least one in which the inequality is strict. This means that:

$\sum C_j(\pi) + \sum V_j(\pi) + E_{max}(\pi) \leq \sum C_j(\beta) + \sum V_j(\beta) + E_{max}(\beta)$, then π is a schedule that gives the best solution than β for (SP), but β is an efficient schedule, and that is a contradiction with our assumption, then β must give an efficient solution for $(TCTVE)$-problem .

2.5. Special cases for problems $(TCTVE)$ and (SP)

Some special cases of problems $(TCTVE)$ and (SP) in this section results in efficient and optimal solutions respectively are introduced.

2.5.1. Special cases for $(TCTVE)$

This part studies various special cases of the $(TCTVE)$ the problem that must have an efficient solution:

Case (2.5.1.1): If $p_1 = d_1$ and $p_j = d_j - d_{j-1}, \forall j, (j = 2,3, \dots, n)$, then SPT schedule σ gives an efficient solution for the problem $(TCTVE)$.

Proof: Since $p_{\sigma_1} = d_{\sigma_1}$ this mean $C_1 = p_{\sigma_1} = d_{\sigma_1}$ but $p_{\sigma_2} = d_{\sigma_2} - d_{\sigma_1} = d_{\sigma_2} - p_{\sigma_1}$ and $C_2 = p_{\sigma_1} + p_{\sigma_2} = p_{\sigma_1} + d_{\sigma_2} - p_{\sigma_1} = d_2$ then $C_2 = d_{\sigma_2}$ and so on $C_j = d_{\sigma_j}$ for $j = 1,2,..,n$. Since $C_j = d_{\sigma_j}$ $\forall j \in \sigma$, this means there are no late and early jobs s.t. $E_j = T_j = 0$ then $E_{max} = V_j = \sum_{j=1}^{n} V_j = 0$ in σ, thus the problem *1 //* $F\left(\sum C_j, \sum V_j, E_{max}\right)$ lowered to *1 //* $\sum C_j$.

However, the SPT rule solves this problem, Hence σ gives an efficient solution for the problem $(TCTVE)$ provided that $p_1 = d_1$ and $p_j = d_j - d_{j-1}, \forall j, (j = 2,3, \dots, n)$.

Case (2.5.1.2): Any schedule α gives *EFSO* for $(TCTVE)$, if $C_j = d_j$ and $p_j = p$ $\forall j$ in α .

Proof: Since $d_j = jp = C_j$, for all j in α this means there are no tardiness and earliness jobs s.t. $E_j = T_j = 0, \forall j \in \sigma$ then $E_{max} = \sum V_j = 0$. Then the problem *1//* $F\left(\sum C_j, \sum V_j, E_{max}\right)$ reduced to *1//* $\sum C_j$. But $\sum_{j=1}^{n} C_j = \sum_{j=1}^{n} jp = p\left(\frac{n^2+n}{2}\right)$ which is constant, hence any schedule gives an efficient solution for $(TCTVE)$.

Case (2.5.1.3): If $kp_j = d_j$ for all $2 \leq k$ then the SPT schedule is an *EFSO* for $(TCTVE)$.

Proof: Since $d_j = kp_j$ then $s_j = kp_j - p_j = (k-1)p_j$, it will be the slack time for the job j, $(j = 1, \dots, n)$. Since the SPT schedule gives the jobs are processed in non-decreasing order.

i.e. $p_1 \leq p_2 \leq \cdots \leq p_n$. Since $(1-k)$ is a positive constant, then $(1-k)p_1 \geq (1-k)p_2 \geq \cdots \geq (1-k)p_n$, then $s_1 \leq s_2 \leq \cdots \leq s_n$. which is MST order, since MST order gives efficient value for the maximum earliness. Hence SPT is efficient for $(TCTVE)$.

Case (2.5.1.4): If $C_j \leq d_{\sigma_j}$ $\forall j$, then sequence $\sigma = SPT = MST$ gives an efficient solution for $(TCTVE)$.

Proof: Since $C_j \leq d_{\sigma_j}$ for all j, this means all jobs are early $\text{s.t.}\, T_j = V_j = \sum V_j = 0$ for all j, hence problem $1 // \left(\sum C_j, \sum V_j, E_{max}\right)$ reduced to *1//* $(\sum C_j, E_{max})$, then σ gives an efficient solution for $(TCTVE)$ since the SPT rule minimum $\sum C_j$ and MST rule minimum E_{max} .

Case (2.5.1.5): If $p_1 \leq \ldots \leq p_n$ and $s_1 \leq \ldots \leq s_n$ then EDD schedule α gives an efficient solution for $(TCTVE)$.

Proof: Since $p_1 \leq \cdots \leq p_n$(which is SPT order) then $\sum_{j=1}^{n} C_j$ is the minimum value, and at the same time $s_1 \leq \cdots \leq s_n$ (which is MST order) hence E_{max} is minimum. But $s_j = d_j - p_j$ and $d_1 - p_1 \leq \ldots \leq d_n - p_n$ then $d_1 - p_1 + p_1 \leq \ldots \leq d_n - p_n + p_n$ (since $p_1 \leq \cdots \leq p_n$), hence $d_1 \leq \ldots \leq d_n$ (which is EDD order), since EDD order gives efficient value for the $\sum_{j=1}^{n} T_j$ then $\sum_{j=1}^{n} V_j$ are minimum. Hence α an efficient solution for the third criterion $\sum C_j$, $\sum V_j$, E_{max} .

Case (2.5.1.6): If $p_j = p$ for all j then the sequence obtained by MST rule gives an *EFSO* for $(TCTVE)$.

Proof: Since $p = p_j$ and $C_j = jp \;\; \forall j$, hence $\sum C_j = p\left(\frac{n^2+n}{2}\right)$, $V_j = min\{T_j, p_j\} =$

$min\{max\{L_j, 0\}, p\} = min\{max\{jp - d_j, 0\}, p\}$ and $E_j = max\{-L_j, 0\} = max\{d_j - jp, 0\}$.

So, there are two cases:

a) If $d_j = jp = C_j, \forall j$ (this means there are no tardy jobs and early jobs $s.t. T_j = E_j = 0, \forall j \in \sigma$) then $V_j = 0$ since $T_j = 0$. This problem $1 //\left(\sum C_j, \sum V_j, E_{max}\right)$ reduced to *1//* $\sum C_j$ but $\sum C_j = p\left(\frac{n(n+1)}{2}\right)$, which is constant this means all schedules give a uniquely efficient solution for any schedule σ.

b) If $d_j > jp = C_j$ (this means all jobs are early s.t. $T_j = 0$ for all j , then $V_j = 0$ and $E_j = max\{-L_j, 0\}$ where $-L_j = d_j - jp, E_{max} = max\{d_j - jp\}$. Hence problem $1//\left(\sum C_j, \sum V_j, E_{max}\right)$ reduced to $1// \left(\sum C_j, E_{max}\right) = 1// \left(p\left(\frac{n^2+n}{2}\right), max\{d_j - jp\}\right)$. Then there is an *EFSQ* in the bi-criteria which fulfills MST rule.

Case (2.5.1.7): If $d_j = d \;\forall\; j$ then the sequence obtained by the SPT rule gives an *EFSO* for the problem $(TCTVE)$.

Proof: For all j, $V_j = min\{T_j, P_j\} = min\{max\{L_j, 0\}, p_j\} = min\{max\{C_j - d, 0\}, p_j\}$

and $E_j = max\{d - C_j, 0\}$, since $d_j = d$. So, there are two cases:

a) If $d_j = d = p_j$ then $C_j = d\left(\frac{n(n+1)}{2}\right)$ and $d \leq C_j, \forall\, j$ (this means all jobs are late $\text{s.t.}\, E_j = 0 \;\forall j$) and $V_j = min\{max\{jd - d, 0\}, d\}$, hence $\sum_{j=1}^{n} V_j = \sum_{j=1}^{n} p_j - d = d(n-1)$. The problem $1 //$ $\left(\sum C_j, \sum V_j, E_{max}\right)$ reduced to *1//* $\left(\sum C_j, \sum V_j\right) = \left(d\left(\frac{n^2+n}{2}\right), nd - d\right)$, which is constant this means all solutions an efficient solutions for any schedule σ.

b) If $d_j = d > p_j$ for all j then (1) If $d > C_j$ (i.e., all jobs are early $\text{s.t.}\, T_j = V_j = \sum V_j = 0$),

$E_j = max\{0, -C_j + d\}, E_{max} = max\{-C_j + d\} = d - p_1$, then the problem *1//* $\left(\sum C_j, \sum V_j, E_{max}\right)$ reduced to *1//* $(\sum C_j, E_{max})$, then there is an *EFSQ* in bi-criteria that fulfills the SPT rule. (2) If $d < C_j$ (this means all jobs are late $\text{s.t.}\, E_j = 0$ for all j) and $V_j = min\{T_j, p_j\} = min\{max\{C_j - d, 0\}, p_j\}$, then problem $1 // \left(\sum C_j, \sum V_j, E_{max}\right)$ reduced to *1//* $\left(\sum C_j, \sum V_j\right) = (\sum C_j, \sum p_j - d)$, then there is an *EFSQ* in bi-criteria this complies with the SPT rule.

Case (2.5.1.8): If $p_j = p$ and $d_j = d$ where $d \geq p$, $\forall j$ in the schedule α, then any schedule α gives an $EFSQ$ for $(TCTVE)$.

Proof: Since all processing times are identical for all j in α, and the due date for all jobs is also identical (i.e., $p_j = p$ and $d_j = d \; \forall j$) then $\sum_{j=1}^{n} C_j = p\left(\frac{n^2+n}{2}\right)$, $E_j = max\{-L_j, 0\} = max\{d - jp, 0\}$, hence $E_{max} = max\{d - jp, 0\} = d - p$ and $V_j = max\{L_j, p\} = max\{max\{jp - d, 0\}, p\}$, thus $\sum V_j = \sum p - d = np - d$. The problem $1 // \left(\sum C_j , \sum V_j , E_{max}\right) = 1// \left(p\left(\frac{n^2+n}{2}\right), np - d, d - p\right)$. Then any schedule is an efficient solution for the problem *1 // F*$\left(\sum C_j , \sum V_j , E_{max}\right)$ because the three quantities are constant.

Case (2.5.1.9): If the three schedules $\sigma = SPT = EDD = MST$, then this schedule gives a unique efficient solution for $(TCTVE)$.

Proof: Since E_{max} minimized by MST rule and since SPT gives $E_{max}\,(\sigma) = E_{max}(MST), T_{max}$ minimized by the EDD rule (well-known that T_{max} is a lower bound for $\sum_{j=1}^{n} V_{\sigma_j}$, i.e., $T_{max}(EDD) \leq \sum_{j=1}^{n} V_{\sigma_j}$), hence if $\sum_{j=1}^{n} T_j$ is minimum, thus minimum $\sum_{j=1}^{n} V_{\sigma_j}$. Then SPT schedule is efficient for the third criterion and hence SPT is efficient for the problem. To prove the uniqueness of . Let π be any schedule, then $\sum C_j\,(\sigma = SPT) \leq \sum C_j\,(\pi)$ and $E_{max}(\sigma = MST) \leq E_{max}(\pi)$ and since T_{max} is lower bound for $\sum V_j$, then $T_{max}(\sigma = EDD) \leq \sum V_j\,(\sigma) \leq \sum V_j\,(\pi)$, thus the solution

$\left(\sum C_j\,(\sigma), \sum V_j\,(\sigma), E_{max}(\sigma)\right)$ dominates the solution $\left(\sum C_j\,(\pi), \sum V_j\,(\pi), E_{max}(\pi)\right)$.

2.5.2. Special cases for sub-problem (SP)

This part studies various special cases of the (SP) the problem that must has an optimal solutions:

Case (2.5.2.1): If $p_1 = d_1 and\ p_j = d_j - d_{j-1}$, $\forall j, (j = 2,3, \ldots, n)$, then SPT schedule σ gives an optimal solution to the 1 // $\sum C_j + \sum V_j + E_{max}$ problem.

Proof: The proof as in case (2.5.1.1) and the problem *1 //*$\sum C_j + \sum V_j + E_{max} = \sum C_j$.Hence $\sigma =$ SPT schedule is an optimal efficient for *1 //* $\sum C_j + \sum V_j + E_{max}$.

Case (2.5.2.2): If $p_j = p\ and\ d_j = C_j$ for all *j* in schedule α then any schedule α gives an *OPSO* for (SP).

Proof: The proof as in case (2.5.1.2) and the problem *1 //* $\sum C_j + \sum V_j + E_{max} = \sum C_j$, $\sum_{j=1}^{n} C_j = p + 2p + 3p + .. + np = p\left(\frac{n^2+n}{2}\right)$. But $p\left(\frac{n^2+n}{2}\right)$ is constant, Hence SRT gives an *OPSO* for (SP) .

Case (2.5.2.3): If $kp_j = d_j$ for all $k \geq 2$ then SPT schedule is an optimal solution for (SP).

Proof: The proof as in case (2.5.1.3) .

Case (2.5.2.4): If $C_j \leq d_{\sigma_j}\ \forall j$, then σ with SPT and MST are (identical sequences) which gives an *OPSO* for (SP).

Proof: The proof as in case (2.5.1.4) and the problem *1 //* $\sum C_j + \sum V_j + E_{max}$ reduced to *1//* $(\sum C_j + E_{max})$. Then σ gives an *OPSO* for (SP) .

Case (2.5.2.5): If $p_1 \leq \ldots \leq p_n$ and $s_1 \leq \ldots \leq s_n$ then EDD schedule α gives an *OPSO* for (SP).

Proof: The proof as in case (2.5.1.5) .

Case (2.5.2.6): If $p_j = p$ for all *j* then the sequence obtained by MST rule gives an *OPSO* for (SP).

Proof: The proof as in case (2.5.1.6) and

$$\sum C_j + \sum V_j + E_{max} = \begin{cases} p\left(\frac{n^2+n}{2}\right), \text{if } d_j = jp = C_j \text{ then } V_j = E_j = 0 \\ p\left(\frac{n^2+n}{2}\right) + max\{d_j - jp\}, \text{if } d_j > jp = C_j \text{ then } T_j = 0 = V_j \end{cases}.$$

Case (2.5.2.7): If $d_j = d$ for all j then the sequence obtained by SPT rule gives an *OPSO* for (SP).

Proof: The proof as in case (5.1.7) and

$$\sum C_j + \sum V_j + E_{max} = \begin{cases} d\left(\frac{n^2+n}{2}\right) + nd - d = d\left(\frac{n^2+3n-2}{2}\right), \text{if } d = p_j \text{ then } d \le C_j, \sum V_j = \sum p_j - d, E_j = 0 \\ \sum C_j + \sum p_j - d + d - p_1 = \sum C_j + \sum p_j - p_1, \text{if } d > p_j \text{ then } \sum V_j = \sum p_j - d, E_{max} = d - p_1 \end{cases}.$$

Case (2.5.2.8): If $p_j = p$, $d_j = d$ for all j in the schedule α, then any schedule α given an *OPSO* for (SP) (where α is any schedule).

Proof: The proof as in case (2.5.1.8) and $\sum C_j + \sum V_j + E_{max} = p\left(\frac{n^2+n}{2}\right) + np - d - p + d = p\left(\frac{n^2+3n-2}{2}\right)$.

Case (2.5.2.9): If there are three schedules (SPT, EDD, and MST are identical), then this schedule gives an *OPSO* for (SP).

Proof: The proof as in case (2.5.1.9) .

By computing, the objective functions (F_{CVE}) and (F_{SP}), respectively, Table 2 gives examples illustrating the special cases (2.5.1) and (2.5.2) of the $(TCTVE)$ and (SP) problems respectively, using $n = 6$.

Table 2. Example of $(TCTVE)'s$ and $(SP)'s$ special cases.

Case	p_j & d_j	Stipulations (Conditions)	F_{CVE}	F_{SP}
(2.5.1.1)	$p_j = 3,6,7,7,8,8$ and $d_j = 3,9,16,23,31,39$.	$p_1 = d_1$ and $p_j = d_j - d_{j-1}$, for $j = 2,..,n$	(121,0,0)	121
(2.5.2.1)	$p_j = 1,2,2,4,4,5$ and $d_j = 1,3,5,9,13,18$.		(49,0,0)	49
(2.5.1.2) (2.5.2.2)	$p_j = 3$ and $d_j = 3,6,9,12,18,21$.	$p_j = p$ and $d_j = jp$, $\forall j$	(69,0,0)	69
(2.5.1.3) (2.5.2.3)	$p_j = 2,1,3,2,1,3$ and $d_j = 4,2,6,4,2,6$	$d_j = kp_j$, $\forall j$	(34,8,1)	43
(2.5.1.4) (2.5.2.4)	$p_j = 4,3,2,2,1,1$ and $d_j = 14,9,5,6,2,3$	$C_j \le d_j$, $\forall j$	(35,0,4)	39
(2.5.1.5) (2.5.2.5)	$p_j = 6,10,12,14,14,18$ and $s_j = 2,4,8,10,12,13$.hence $d_j = 8,14,20,24,26,31$.	$p_i \le p_j$ and $s_i \le s_j$ for all j .	(222,56,2)	280
(2.5.1.6)	$p_j = 3$ and $d_j = 3,4,6,7,8,9$ and $C_j \ge d_j$		(63,14,0)	77
(2.5.2.6)	$p_j = 2$ and $d_j = 4,6,8,10,12,14$ and $d_j \ge C_j$.	$p_j = p, \forall j$	(42,0,12)	54
(2.5.1.7)	$p_j = 3$ and $d_j = 3$ and $d \le C_j$ for all j.		(45,12,0)	57

(2.5.2.7)	$p_j = 5,4,3,2,1,7$ and $d_j = 30$ and $d > C_j$.	$d_j = d, \forall j$	(57,0,29)	86
(2.5.1.8)	$p = 5$ and $7 = d$ and $d > p$.	$p_j = p$, $d_j = d$ for all j	(105,18,2)	125
(2.5.2.8)	$p = 5 = d$.		(105,25,0)	130
(2.5.1.9)	$p_j = 2,3,5,8,9,9$ and	$SPT = EDD = MST$ for all j	(98,0,0)	98
(2.5.2.9)	$d_j = 2,5,10,18,27,36$.			

Where F_{CVE} **:** is the multi-criteria of problem $(TCTVE)$**,** F_{SP} **:** is the multi-objective function of problem (SP)**.**

3. Results and discussion

In this section, the following results are formed in the light of the previous theories, propositions, and some cases based on them:

- The SPT rule given an efficient solution for the problem $1//F(\sum C_j, \sum V_j, E_{max})$ and optimal solution for the problem $//\sum C_j + \sum V_j + E_{max}$,this proved in proposition (1).
- Every optimal solution for the problem $1//\sum C_j + \sum V_j + E_{max}$ is an efficient solution to the problem $1//F(\sum C_j, \sum V_j, E_{max})$, this proved in proposition (2).
- The SPT schedule σ gives an efficient solution for problem $(TCTVE)$ and optimal solution for problem $1//\sum C_j + \sum V_j + E_{max}$, when one of the following conditions is fulfilled: 1) $p_1 = d_1$ and $p_j = d_j - d_{j-1}$, $\forall j, (j = 2,3, \ldots, n)$ 2) $kp_j = d_j$ for all $k \geq 2$ 3) $d_j = d \; \forall \; j$.
- Any schedule α given an efficient solution for problem $(TCTVE)$ and optimal solution for problem $1//\sum C_j + \sum V_j + E_{max}$ when $p_j = p$, $d_j = d$ for all j in the schedule α.

4. Conclusions and future works

In this study, a mathematical model was created to address the research problems *1//*$F(\sum C_j, \sum V_j, E_{max})$, *1//* $\sum C_j + \sum V_j + E_{max}$ and it has been proven that certain rules provide efficient (optimal) solutions to the $(TCTVE)$ and (SP) problems, finding and proving some certain cases that discover some efficient (optimal) solutions for $(TCTVE)$ and (SP)problem under consideration and demonstrating that SPT and EDD give efficient (optimal) solutions to these problems, demonstrated the significance of the Dominance Rule (DR) that can be used in this problem to improve efficient solutions.

In the future, interesting would be to conduct research on the following MSPs.

1) $1/r_j/F(\sum C_j, \sum V_j, E_{max})$.
2) $1/r_j/\sum C_j + \sum V_j + E_{max}$.
3) $1/S_f/F(\sum C_j, \sum V_j, E_{max})$.

Acknowledgments: The authors of this paper would like to thank the referees for their helpful suggestions and comments.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

References

[1] F. H. Ali and M. G. Ahmed, “Local Search Methods for Solving Total Completion Times, Range of Lateness and Maximum Tardiness Problem,” *Proc. 6th Int. Eng. Conf. ’ ’Sustainable Technol. Dev. IEC 2020*, no. 1, pp. 103–108, 2020.

[2] C. N. Potts and L. N. van Wassenhove, “Single Machine Scheduling to Minimize Total Late Work,” *Oper. Res.*, vol. 40, no. 3, pp. 586–595, 1992.

[3] H. Hoogeveen, “Multicriteria scheduling,” *Eur. J. Oper. Res.*, vol. 167, no. 3, pp. 592–623, 2005.

[4] M. Pinedo, “Scheduling : Theory, Algorithms and System Dvelopment,” *Oper. Res. Proc. 1991*, no. March, pp. 35–42, 1992.

[5] F. S. Fandi, “Minimizing total Completion Time and Maximum late Work Simultaneously Notation and Basic Concepts :,” *Ibn Al-Haitham J. Pure Appl. Sci.*, vol. 25, no. 3, pp. 356–366, 2012.

[6] F. H. Ali, R. N. Jawad, and W. A. Hussein, “Solving Bi-Criteria and Bi-Objectives of Total Tardiness Jobs Times and Range of Lateness Problems Using New Techniques,” *Al-Mustansiriyah J. Sci.*, vol. 33, no. 3, pp. 27–35, 2022.

[7] M. H. Ibrahim, “Exact and Approximation Methods for Solving Machine Scheduling Problems with and without Setup Time,” Mustansiriyah University, 2022.

[8] D. A. Hassan, N. Mehdavi-Amiri, and A. M. Ramadan, “A heuristic approach to minimize three criteria using efficient solutions,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 6, no. 1, pp. 334–341, 2022..

[9] W. E. Smith, “Various optimizers for single-stage production,” *Nav. Res. Logist. Q.*, vol. 3, no. 1–2, pp. 59–66, 1956.

[10] J. A. Hoogeveen, “Single-Machine Scheduling to Minimize a Function of Two or Three Maximum Cost Criteria,” *J. Algorithms*, vol. 21, no. 2, pp. 415–433, 1996.

[11] J. A. Hoogeveen, “Minimizing maximum promptness and maximum lateness on a single machine,” *Math. Oper. Res.*, vol. 21, no. 1, pp. 100–114, 1996.

[12] A. A. Jawad, F. H. Ali, and W. S. Hasanain, “Using heuristic and branch and bound methods to solve a multi-criteria machine scheduling problem,” *Iraqi J. Sci.*, vol. 61, no. 8, pp. 2055–2069, 2020.

[13] B. A. Amin and A. M. Ramadan, “Novel Heuristic Approach for Solving Multi-objective Scheduling Problems,” *Ibn AL- Haitham J. Pure Appl. Sci.*, vol. 34, no. 3, pp. 50–59, 2021.

[14] F. H. Ali and A. A. Jawad, “Minimizing the total completion time and total earliness time functions for a machine scheduling problem using local search methods,” *Iraqi J. Sci.*, vol. 2020, pp. 126–133, 2020.

[15] B. Atiya, A. J. K. Bakheet, I. T. Abbas, M. R. A. Bakar, L. L. Soon, and M. B. Monsi, "Application of simulated annealing to solve multi-objectives for aggregate production planning". In AIP Conference Proceedings, Vol. 1739, No. 1, p. 020086), 2016.

[16] F.H. Ali, M.G. Ahmed, "Optimal and near Optimal Solutions for Multi Objective Function on a Single Machine", *Proc. 2020 Int. Conf. Comput. Sci. Softw. Eng. CSASE*, pp.152–156, 2020.