DOI: 10.24996/ijs.2024.65.5.30

ISSN: 0067-2904

Solving the Multi-Criteria Problem: Total Completion Time, Total Late Work, Total Earliness Time, Maximum Earliness, and Maximum Tardiness

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

Received: 7/3/2023 Accepted: 3/5/2023 Published: 30/5/2024

Abstract:
In this paper, we study the scheduling of n jobs on a single machine. Each of the n jobs will be processed without interruption and becomes available for processing at time zero. The goal is to find a processing order for the jobs, minimizing the total completion time, total late work, total earliness time, and maximum earliness maximum tardiness. The posed problems in this paper are as follows: The first problem is to minimize the multi-criteria, which includes minimizing the total completion time, total late work, total earliness time, maximum earliness, and maximum tardiness that are denoted by $\sum C_j, \sum V_j, \sum E_j, E_{max}, and\ T_{max}$, respectively. The second problem is to minimize the multi-objective functions $(\sum C_j + \sum V_j, + \sum E_j + E_{max} + T_{max})$. The theoretical section will present the mathematical formula for the discussed problem. Because these problems are NP-hard problems. It is difficult to determine the efficient (optimal) solution set for these problems. Some special cases are shown and proven to find efficient (optimal) solutions to the discussed problem. The significance of the dominance rule can be applied to problems to improve and to get good solutions that will be highlighted.

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

حل مشكلة متعددة المعايير: تقليل مجموع وقت اتمام العمل، و مجموع العمل المتأخر، و مجموع وقت التبكير عن بدا العمل، و تقليل الحد الاعلى لوقت التبكير و الحد الاعلى لوقت التأخير عن العمل

نغم موسى نعمه 1و2، بيداءعطية 2
[1]القسم الرياضيات، الكلية العلوم للبنات، الجامعة بغداد، بغداد، العراق
[2]القسم الرياضيات، الكلية التربية للعلوم الصرفة ابن الهيثم، الجامعة بغداد، بغداد، العراق

الخلاصه:
هذا البحث يدرس جدولة عدد من الاعمال على الماكنة المنفردة. هدفنا العثور علي تسلسل لمعالجة الاعمال لتقليل مجموع وقت اتمام العمل، و مجموع العمل المتأخر، و مجموع وقت التبكير عن بدا العمل، و تقليل الحد الاعلى لوقت التبكير و الحد الاعلى لوقت التأخير عن العمل يتم معالجة كل عمل دون انقطاع

* Email: nagham2182011@gmail.com

جميع الاعمال متاحة للمعالجة عن الوقت صفر. هذا البحث سيتناول دراسة مسالة تقليل المعايير المتعددة والتي تشمل مجموع وقت اتمام العمل، و مجموع العمل المتأخر، و مجموع وقت التبكير عن بدا العمل، و تقليل الحد الاعلى لوقت التبكير و الحد الاعلى لوقت التأخير عن العمل $(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max})$ و تقليل مجموع اجمالي وقت اتمام العمل، و مجموع العمل المتأخر، و مجموع وقت التبكير عن بدا العمل، و تقليل الحد الاعلى لوقت التبكير و الحد الاعلى لوقت التأخير عن العمل $(\sum C_j + \sum V_j, +\sum E_j + E_{max} + T_{max})$. الجزء النظري في هذه الورقة يقدم الصيغة الرياضية للمسالة التي سيتم مناقشتها، و يتضمن اثبات بعض الحالات الخاصة التي تساعد على ايجاد الحلول الفعالة (المثلى) المناسبة لها، نظرًا لأن مسالتنا تعد من المسائل الصعبة NP-hard ، لذا من الصعب تحديد مجموعة كل الحلول الفعالة (المثلى) لهذه المسائل، و بالإضافة الى تسليط الضوء على قواعد الهيمنة (DR) لتقليل عدد التسلسلات التي يمكن تطبيقها على هذه المسالة لتحسين الحلول الفعالة.

1. Introduction

Since 1954, scheduling problems have received much attention in the literature. Initially, the researchers looked at only one objective function[1]. In practical cases, the decision-maker is bound to choose only one of some objectives. Nowadays, research on multi-criteria scheduling problems has increased. Nagar et al [2] presented a survey of multiple and binary problems in scheduling. In general, there are two structures for dealing with conflicting criteria, namely hierarchical minification and concurrent minification [3]. The first one is the primary criterion, and the other is the secondary criterion. In this case, one reduces the primary criterion and chooses a table with a minimum value for the second criterion. In the second approach, the effective solutions (Pareto set) will be generated, and the decision maker is the one with the best composite objective function [4]. The first paper on a problem of this kind was presented by Smith [5]. In this work, the problem of scheduling n jobs on a single machine can be dealt with at most one job at a time without interruption. Each job becomes available for processing at time zero, which requires a positive processing time.

In general, scheduling means allocating machines to jobs in order to complete all jobs under imposed constraints. Problem with scheduling an N={1,…,n} group of jobs on a single machine. Each job $j, j \in N$ has an integer processed time p_j, the due date d_j. Given the schedule $\rho = (\rho(1), \rho(2), \dots, \rho(n))$, then for each job j, we calculate the completion time by $C_1 = p_1$ and $C_j = \sum_{k=1}^{n} p_{\rho_k}$ for $j = 2,3, \dots, n$. The earliness of the job j is defined by $E_j = max\left\{d_{\rho_j} - C_j, 0\right\}$, the tardiness of the job j is defined by $T_j = max\left\{C_j - d_{\rho_j}, 0\right\}$ and the Late work is defined by $V_j = min\left\{T_j, p_{\rho_j}\right\}$. So, there is a total completion time $\sum_{j \in N} C_j$, total Late work $\sum_{j \in N} V_j$, maximum earliness $E_{max} = max_{j \in N}\{E_{max}\}$, and maximum tardiness $T_{max} = max_{j \in N}\{T_{max}\}$. The total completion time of 1// $\sum C_j$ problem is minimized by the short processing time (SPT) rule which is optimal for Smith 1956 [5][6]. The maximum earliness for the 1// $\sum E_j$ problem is minimized by the minimum slack time(MST) rule [4][6]. The maximum tardiness for 1// T_{max} problem is minimized by the earliest due date (EDD) rule to Jackson 1955 [2][6], the two problems 1// $\sum E_j$, 1// $\sum V_j$, and 1// $\sum T_j$ are NP-hard [6],[3],[7],[8],[9],[10]. Any problem including cost functions as sub-problems is NP-hard. Any problem including cost functions as sub-problems is NP-hard.

The most important literature survey for the last eight years. Z. M. Ali and T. S. Abdul Razaq 2015 [9] discussed the multi-criteria in order to establish a collection of efficient solutions for the general problem, and scheduling problems that are researched on a single machine are considered. *1//* $F(\sum C_j, \sum T_j, T_{Max})$, *1//* $F(\sum C_j, \sum E_j, E_{Max})$, *1//* $\sum C_j + \sum T_j + T_{Max}$, *1//* $\sum C_j + \sum E_j + T_{Max}$. M. G. Ahmed and F. H. Ali 2022[11]examined the multi-objective problem, which is the sum of completion time, tardiness, earliness, and late work. $1//\sum_{=1}^{n}(E_j + T_j + C_j + U_j + V_j)$, $1//\sum_{=1}^{n}(\alpha_j E_j + \beta_j T_j + \theta_j C_j + \gamma_j U_j + \omega_j V_j)$, $1/S_f/\sum_{=1}^{n}(\alpha_{jf} E_{jf} + \beta_{jf} T_{jf} + \theta_{jf} C_{jf} + \gamma_{jf} U_{jf} + \omega_{jf} V_{jf})$. They suggested an Upper Bound (limits) UB and a Lower Bound (limits) LB to be used in the application of the Branch and Bound method. F. H. Ali and M. G. Ahmed 2022 [12] studied the multi-criteria$(\sum \boldsymbol{C_j}, \boldsymbol{T_{max}}, \boldsymbol{R_L})$, multi-objective function $(\sum \boldsymbol{C_j} + \boldsymbol{T_{max}} + \boldsymbol{R_L})$and found the optimal solution by using the Branch and Bound method with and without DR then they used some heuristic methods. D. A. Hassan, N. Mehdavi-Amiri, and A. M. Ramadan 2022[13] introduced a heuristic algorithm to reduce the $(\sum \boldsymbol{C_j} + \boldsymbol{E_{max}} + \boldsymbol{T_{max}})$ in single-machine scheduling.

This paper displays multi-criteria scheduling problems and begins with some basic scheduling concepts of the multi-criteria problem. Basic rules are given in Section (1). In Section (2), the mathematical formula for the discussed problem will be presented and provided information on the formulation and analysis of the problem. In Section (3), some special cases are shown and proven which find some efficient (optimal) and suitable solutions to the discussed problem. We also show there exists an effective solution to problems and prove several rules. The Dominance Rule is described in Section (4). In Section (5), the significant obtained results in the previous section are presented and discussed. The conclusions and lists of future works are given in Section (6).
This paper uses some important rules and definitions:

Shortest Processing Tim (*SPT*)*:* Jobs are sequenced in non-decreasing order of the processing times p_j (i.e. $p_1 \le p_2 \le \cdots \le p_n$), this rule is well-known to minimize $\sum C_j$ for problem $1//\sum C_j$ [5].

Earliest Due Date (*EDD*)*:* Jobs are sequenced in non-decreasing order of their due dates d_j(i.e. $d_1 \le d_2 \le \cdots \le d_n$), this rule is used to minimize T_{max} for problem $1//T_{max}$ [14].
Minimum Slack Time (*MST)*: Jobs are sequenced in non-decreasing order of their slack time $s_j = d_j - p_j$ (i.e. $s_1 \le s_2 \le \cdots \le s_n$). To minimize E_{max} by using this rule [4].
Efficient Solution: A schedule α^* is known as an efficient solution or the Pareto optimal or (non-dominated) If we cannot find another schedule α that satisfies $h_j(\alpha) \le h_j(\alpha^*), j = 1,2,..,n$ with at least one of the above considered a strict disparity. Another way is α^* which is dominated by α [13][6].

Definition: The σ^* the schedule is considered to be optimal if there is no other schedule σ satisfies $f_j(\sigma) \le f_j(\sigma^*), j = 1, \dots, k$ (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^*[16][15] .

2. Description of Multi-Criteria Scheduling Problem

In this section, the five-criteria scheduling problems to be studied will be described. Let the number of jobs available at time 0 that is 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, a sequence of jobs $\rho = (\rho_1, \rho_2, \dots, \rho_n)$ is given, the earliest completion time $C_j = \sum_{k=1}^{j} p_{\rho_k}$is generated, the $T_j = max\left\{C_j - d_{\rho_j}, 0\right\}$ job j's tardiness, the earliness of job j, $E_j = max\left\{d_{\rho_j} - C_j, 0\right\}$, the tardiness of job j, $T_j = max\left\{C_j - d_{\rho_j}, 0\right\}$, and $V_j = min\left\{T_j, p_{\rho_j}\right\}$ the job j's late work. The aim of this problem is to find a schedule $\sigma \in \mathcal{S}$, where $\mathcal{S}$ is the set of all possible feasible schedules that minimize the quintet criteria $\left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$, which is denoted by $\left(S_{(CVE)}M_{ET}\right)$, it can be mathematically formulated as follows:

$F_{S_{(CVE)}M_{ET}} = Min\left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$.

Subject to

$$\left.\begin{array}{ll} C_j \geq p_{\rho_j}, & j = 1, \dots, n \\ C_j = \sum_{k=1}^{j-1} p_{\rho k} + p_{\rho j}, & j = 1,2, \dots, n \\ T_j \geq C_j - d_{\rho_j}, & j = 1, \dots, n \\ E_j \geq d_{\rho_j} - C_j, & j = 1, \dots, n \\ V_j = min\left\{T_j, p_{\rho_j}\right\}, & j = 1,2, \dots, n \\ V_j \geq 0, \quad E_j \geq 0, \text{and } T_j \geq 0, & j = 1, \dots, n \end{array}\right\} \quad \left(S_{(CVE)}M_{ET}\right).$$

The ρ_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 $\left(S_{(CVE)}M_{ET}\right)$ is challenging, since it is an NP-hard problem because the problems *1//* $\sum_{j=1}^{n} V_j$, *1//* $\sum_{j=1}^{n} E_j$ are NP-hard [7][12].

Proposition (1): There is an efficient sequence for problem $1//F\left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$ that satisfies the short processing time rule.

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

$\sum C_j(\delta) \leq \sum C_j\,(SPT^*), \sum V_j\,(\delta) \leq \sum V_j\,(SPT^*), \sum E_j\,(\delta) \leq \sum E_j\,(SPT^*),\ E_{max}(\delta) \leq E_{max}(SPT^*)$, and $T_{max}(\delta) \leq T_{max}(SPT^*)$ (1).

The presence of at least one of the strict inequalities.

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

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. This shows 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

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

Hence, each one of the SPT^* sequences is efficient as a result of the EDD and MST rules .

As mentioned in proposition (1), we show that the SPT rule is efficient for the problem $\left(S_{(CVE)}M_{ET}\right)$, however, the next example shows that the EDD rule does not.

Example (1): Suppose the problem $\left(S_{(CVE)}M_{ET}\right)$ has the following data:

	Job1	*Job2*	*Job3*	*Job4*	*Job5*	*Job6*
p_j	*10*	*4*	*3*	*3*	*8*	*7*
d_j	*13*	*10*	*8*	*17*	*10*	*14*

s_j	3	6	5	14	2	7

A feasible schedule is provided by the SPT rule (3,4,2,6,5,1) and (4,3,2,6,5,1), hence $(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}) = (96,21,16,11,22)$ from SPT^* order (3,4,2,6,5,1) and $(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}) = (96,21,16,14,22)$ from SPT order (4,3,2,6,5,1), it is clear that in the SPT^*sequence the tasks (1,3) are arranged with equal processing time in the rule of the MST or EDD. But EDD rule (3,2,5,1,6,4) with $(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}) = (117,25,8,5,18)$ and MST rule (5,1,3,2,6,4) with $(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}) = (139,22,2,2,18)$ hence SPT^* the sequence gives an efficient solution for the problem $(S_{(CVE)}M_{ET})$.

2.1 Description of Multi-Objective Scheduling 1// $\sum C_j + \sum V_j + \sum E_j + E_{max} + T_{max}$ Sub-problem of Problem $(S_{(CVE)}M_{ET})$.

The problem $1//F(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max})$ can deduce a sub-problem, that it minimizes 1// $(\sum C_j + \sum V_j + \sum E_j + E_{max} + T_{max})$. This problem is described as follows:
Assume that ρ is any schedule that can be expressed as follows for a certain schedule $\rho = (\rho_1, \rho_2, \dots, \rho_n)$:

$$\left.\begin{aligned}
&F_{SP} = Min(\textstyle\sum C_j + \sum V_j + \sum E_j + E_{max} + T_{max}) \\
&\qquad \text{subject to} \\
&C_j = \textstyle\sum_{k=1}^{j} p_{\rho_k} && j = 1,2,\dots,n \\
&C_j = C_{(j-1)} + p_{\rho_j} && j = 2,3,\dots,n \\
&E_j \geq d_{\rho_j} - C_j && j = 1,2,\dots,n \\
&T_j \geq C_j - d_{\rho_j} && j = 1,2,\dots,n \\
&V_j = min\{T_j, p_{\rho_j}\} && j = 1,2,..,n \\
&V_j \geq 0,\ E_j \geq 0,\ T_j \geq 0 && j = 1,2,\dots,n
\end{aligned}\right\} \quad (SP).$$

The objective of the NP-hard problem sub-problem 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.

Proposition (2): Each optimal solution for the sub-problem is an efficient solution to the problem $(S_{(CVE)}M_{ET})$.
Proof: let β be an optimal schedule for the sub-problem. Suppose that β gives no efficient solution for the problem $(S_{(CVE)}M_{ET})$, then there is an efficient schedule say π for $(S_{(CVE)}M_{ET})$ the problem such that:
$\sum C_j(\pi) \leq \sum C_j(\beta)$,$\sum V_j(\pi) \leq \sum V_j(\beta)$,$\sum E_j(\pi) \leq \sum E_j(\beta)$, $E_{max}(\pi) \leq E_{max}(\beta)$, and $T_{max}(\pi) \leq T_{max}(\beta)$.
At least one in which the inequality is strict. This means that:
$\sum C_j(\pi) + \sum V_j(\pi) + \sum E_j(\pi) + E_{max}(\pi) + T_{max}(\pi) \leq \sum C_j(\beta) + \sum V_j(\beta) + \sum E_j(\beta) + E_{max}(\beta) + T_{max}(\beta)$, 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 $(S_{(CVE)}M_{ET})$ problem .

3. Special Cases (SC) for problems $(S_{(CVE)}M_{ET})$ and sub-problem.

This part studies various special cases of the $(S_{(CVE)}M_{ET})$ the problem that must have an efficient solution. The special case of the scheduling problem means we obtain an efficient (optimal) schedule (efficient (optimal) solution) directly without using the (BAB) or (DP) method.

3.1 Special Cases for problem$\left(S_{(CVE)}M_{ET}\right)$

This part studies various special cases of the $\left(S_{(CVE)}M_{ET}\right)$ the problem that must have an efficient solution:

Case (3.1.1): If $p_j = p$ and $d_j = jp, \forall j$ in the schedule of ρ, then ρ gives the efficient schedule for the problem $\left(S_{(CVE)}M_{ET}\right)$.

Proof: Since $d_j = C_j$,$\forall j \in \rho$, this means there is no job late and early s.t. $E_j = T_j = V_j = 0$) then $\sum_{j=1}^{n} E_j = \sum_{j=1}^{n} V_j = E_{max} = T_{max} = 0$. Then the problem $1// F\left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$ is reduced to *1//* $\sum_{j=1}^{n} C_j$. 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 $\left(S_{(CVE)}M_{ET}\right)$.

Case (3.1.2): If $p_1 = d_1$ and $p_j = d_j - d_{j-1}$, for all j in ρ (except 1) then the short processing time schedule α gives an efficient schedule for the problem $\left(S_{(CVE)}M_{ET}\right)$.

Proof: Since $p_1 = d_1$ and $p_2 = d_2 - d_1 = d_2 - p_1$, then $C_1 = d_1$ and $C_2 = p_1 + p_2 = p_1 + d_2 - p_1 = d_2$ then $C_2 = d_2$ and so on $C_j = d_j$ for $j = 1,2,..,n$. Since $C_j = d_j$ for all j in σ then $L_j = 0, \forall j$, and $E_j = T_j = V_j = 0$, so $\sum V_j = \sum E_j = E_{max} = T_{max} = 0$. Problem 1 // $\left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$ is reduced to *1 //* $\sum C_j$. But the rule that solved this problem was short processing time. Then ρ provides an efficient solution to $\left(S_{(CVE)}M_{ET}\right)$ problem.

Case (3.1.3): If $p_1 \leq \ldots \leq p_n$ and $s_1 \leq \ldots \leq s_n$ in schedule ρ then schedule ρ gives an efficient solution for $\left(S_{(CVE)}M_{ET}\right)$.

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} and $\sum E_j$ are 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 T_{max} and $\sum_{j=1}^{n} T_j$, then $\sum_{j=1}^{n} V_j$ are minimum. Hence, ρ an efficient solution to the problem $\left(S_{(CVE)}M_{ET}\right)$.

Case (3.1.4): If $C_j \leq d_{\rho_j}$ $\forall j$, then sequence $\rho = SPT = MST$ gives an efficient solution for $\left(S_{(CVE)}M_{ET}\right)$.

Proof: Since $C_j \leq d_{\rho_j}$ for all j, this means all jobs are early s.t. $T_j = V_j = \sum V_j = T_{max} = 0$ for all j, hence problem 1 // $\left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$ reduced to *1 //* $(\sum C_j , \sum E_j, E_{max})$, then σ gives an efficient solution for $\left(S_{(CVE)}M_{ET}\right)$ since the short processing time rule minimizes $\sum C_j$ and MST rule minimizes $\sum E_j, E_{max}$.

Case (3.1.5): If $p_j = p$ and $d_j = d$ where $d \geq p$, $\forall j$ in the schedule ρ, then schedule ρ gives an efficient solution for $\left(S_{(CVE)}M_{ET}\right)$.

Proof: Since all processing times are identical for all j in ρ, 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 $T_j = max\{L_j, 0\} = max\{jp - d, 0\}$ and $T_{max} = max\{jp - d, 0\} = np - d$ and $V_j = min\{L_j, p\} = min\{max\{jp - d, 0\}, p\}$, thus $\sum V_j = \sum p - d = np - d$, and there are two cases for p and d:

a) If $d_j = d = p_j = p$, then $C_j = d\left(\frac{n(n+1)}{2}\right)$ and $d \leq C_j, \forall j$, this means all jobs are late s.t. $E_j = 0, \forall j$) and $V_j = min\{max\{jd - d, 0\}, d\}$, hence $\sum_{j=1}^{n} V_j = \sum_{j=1}^{n} p - d = d(n-1)$.

Problem $1 // \left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$ is reduced to $1// \left(\sum C_j, \sum V_j, T_{max}\right) = \left(d\left(\frac{n^2+n}{2}\right), nd - d, d(n-1)\right)$ which is constant this means all solutions an efficient solutions for any schedule ρ.
b) If $d_j = d > p_j = p$ for all j then (1) If $d > C_j$ that means all jobs are early $\text{s.t.}\, T_j = V_j = \sum V_j = 0$.
$E_j = max\{0, -C_j + d\} = max\{0, -jp + d\}, E_{max} = max\{-jp + d\} = d - p$, then the problem $1// \left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$ is reduced to $1// (\sum C_j, \sum E_j, E_{max})$. (2) If $d < C_j$ (this means all jobs are late such that $E_j = 0$ for all j) and $V_j = min\{T_j, p_j\} = min\{max\{C_j - d, 0\}, p\}$, then problem $1 // \left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$ is reduced to $1// \left(\sum C_j, \sum V_j, T_{max}\right) = (\sum C_j, \sum d - d, nd - d)$. (3) If there are $d < C_j$, $d > C_j$, and $d = C_j$, then problem $1 // \left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right) = 1// \left(p\left(\frac{n^2+n}{2}\right), np - d, d - p, \sum max\{jp - d, 0\}, np - d\right)$. Then any schedule is an efficient solution for problem $1 // F\left(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max}\right)$ because the six quantities are constant.
Case (3.1.6): If the three schedules SPT,EDD, and MST have the same order (schedule) ρ, then this schedule gives an efficient and unique solution for $\left(S_{(CVE)}M_{ET}\right)$.
Proof: Since $\sum E_j, E_{max}$ is minimized by MST rule and since SPT gives $\sum E_j(\rho) = \sum E_j(MST)$, $E_{max}(\rho) = E_{max}(MST)$ and T_{max} is minimized by the EDD rule , it is well-known that T_{max} is a lower bound for $\sum_{j=1}^{n} V_{\rho_j}$, $\text{i.e.}, T_{max}(EDD) \le \sum_{j=1}^{n} V_{\rho_j}$. Hence, if $\sum_{j=1}^{n} T_j$ is minimum, thus minimum $\sum_{j=1}^{n} V_{\rho_j}$. Then SPT schedule is efficient for the third criterion and hence SPT is efficient for the problem.
To prove the uniqueness of ρ, let π be any schedule, then $\sum C_j\,(\rho = SPT) \le \sum C_j\,(\pi)$ and $\sum E_j\,(\rho = MST) \le \sum E_j\,(\pi)$ and $E_{max}(\rho = MST) \le E_{max}(\pi)$ and since T_{max} is the lower bound for $\sum V_j$, then $T_{max}(\rho = EDD) \le \sum V_j\,(\rho) \le \sum V_j\,(\pi)$, thus the solution $\left(\sum C_j(\rho), \sum V_j(\rho), \sum E_j(\rho), E_{max}(\rho), T_{max}(\rho)\right)$ dominates the solution $\left(\sum C_j(\pi), \sum V_j(\pi), \sum E_j(\pi), E_{max}(\pi), T_{max}(\pi)\right)$.

3.2 Special Cases for Sub-problem (SP)
This part studies various special cases of the sub-problem of the problem that must have an optimal solution:
Case (3.2.1): If $p_j = p$ and $d_j = jp, \forall\, j$ in the schedule of ρ, then ρ gives the efficient schedule for the problem (SP) .
Proof: The proof is the same as in the case (3.11).
Case (3.2.2): If $p_1 = d_1$ and $p_j = d_j - d_{j-1}$, for all j in ρ (except 1) then SPT schedule α gives an efficient schedule for the problem (SP) .
Proof: The proof is the same as in the case (3.1.2).

Case (3.2.3): If $p_1 \le \ldots \le p_n$ and $s_1 \le \ldots \le s_n$ in schedule α then schedule ρ gives an efficient solution for the sub-problem .
Proof: The proof is the same as in the case (3.1.3).
Case (3.2.4): If $C_j \le d_{\rho_j}\ \forall j$, then sequence $\rho = SPT = MST$ gives an efficient solution for $\left(S_{(CVE)}M_{ET}\right)$.
Proof: The proof is the same as in the case (3.1.4).

Case (3.2.5): If $p_j = p$ and $d_j = d$ where $d \geq p$, $\forall j$ in the schedule ρ, then schedule ρ gives an $EFSQ$ for the sub-problem.
Proof: The proof is the same as in the case (3.1.5).
Case (3.2.6): If the three schedules SPT,EDD, and MST have the same order (schedule) ρ, then this schedule gives an efficient and unique solution for the sub-problem.
Proof: The proof is the same as in the case (3.1.6).
By computing, the objective functions $\left(F_{S_{(CVE)}M_{ET}}\right)$ and (F_{SP}), respectively. Table 1 gives examples that illustrate the special cases (3.1) and (3.2) of the $\left(S_{(CVE)}M_{ET}\right)$ and (SP) problems with $n = 6$, respectively.

Table 1: Example of $\left(S_{(CVE)}M_{ET}\right)'s$ and $(SP)'s$ special cases

Case	$\boldsymbol{p_j}$ & $\boldsymbol{d_j}$	**Stipulations (Conditions)**	$F_{S_{(CVE)}M_{ET}}$	F_{SP}
(3.1.1) (3.2.1)	$p_j = 2$ and $d_j = 2,4,6,8,10,12$.	$p_j = p$ and $d_j = jp$,$\forall j$	(42,0,0,0,0)	42
	$p_j = 3$ and $d_j = 3,6,9,12,15,18$.		(63,0,0,0,0)	63
(3.1.2)	$p_j = 2,3,4,6,8,10$ and $d_j = 2,5,9,15,23,33$.	$p_1 = d_1$and $p_j = d_j - d_{j-1}$,	(87,0,0,0,0)	87
(3.2.2)	$p_j = 1,2,2,4,4,5$ and $d_j = 1,3,5,9,13,18$.	for $j = 2,..,n$	(49,0,0,0,0)	49
(3.1.3)	$p_j = 6,10,12,8,4,14$ and $s_j = 2,4,4,4,0,4$, hence $d_j = 8,14,16,12,4,18$.	$p_i \leq p_j$ and $s_i \leq s_j$ for all j .	(154,44,0,0,36)	234
(3.2.3)	$p_j = 6,10,12,14,14,18$ and $s_j = 2,4,8,10,12,13$, hence $d_j = 8,14,20,24,26,31$.		(222,56,2,2,43)	325
(3.1.4) (3.2.4)	$p_j = 8,5,2,6,4,3$ and $d_j = 30,16,2,20,12,6$.	$C_j \leq d_j$,$\forall j$	(78,0,8,3,0)	89
	$p_j = 4,3,2,2,1,1$ and $d_j = 14,9,5,6,2,3$		(35,0,4,1,0)	40
(3.1.5)	$p = 4$ and $8 = d$ and $d > p$.	$p_j = p$, $d_j = d$ for all j	(84,16,4,4,16)	124
(3.2.5)	$p = 4 = d$.		(84,20,0,0,20)	124
(3.1.6) (3.2.6)	$p_j = 2,8,4,7,6,3$ and $d_j = 3,17,7,14,10,6$.	$SPT = EDD = MST$ for all j	(83,22,2,1,13)	121
	$p_j = 2,3,5,8,9,9$ and $d_j = 2,5,10,18,27,36$.		(98,0,0,0,0)	98

Where $F_{\left(S_{(CVE)}M_{ET}\right)}$ is the multi-criteria of the problem $\left(S_{(CVE)}M_{ET}\right)$, F_{SP} is the multi-objective function of the problem (SP).

4. Dominance Rules (DRs) for MSP

Dominance rules are most useful when a node in a tree that has a good lower bound can be eliminated which is less than the optimal solution. Dominance rules can also be used using the Branch and Bound procedure to cancel nodes that may be dominated by other nodes. The effects of these improvements will help reduce the number of nodes that are too large for an optimal solution. The sequence discussed is reduced when some control rules (DRs) are used**.**
In this part, we give some important definitions that are used in the remaining of this work:

Definition[17]**:** Graph G represents a finite number of nodes or vertices V and a finite number of edges, connecting two vertices, and the edge connecting the vertex to itself is called a loop.

Definition[17]**:** If n vertices make up a graph called G, then $A(G) = [a_{ij}]$ is the matrix, which is called an 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 , $a_{ij} = \begin{cases} 0, \text{if } i = j \text{ or } i \nrightarrow j \\ 1, \text{if } i \to j \\ a_{ij}, \text{ otherwise.} \end{cases}$.

Theorem: If $p_i \le p_j$ and $d_i \le d_j$ then there is an optimal schedule for the problem (SP) in which job i processing before job .

Proof: Suppose there is a sequence $\sigma = \sigma_1 ij \sigma_2$ and a schedule $\dot{\sigma} = \sigma_1 ji \sigma_2$ which is obtained by interchanging the jobs i and j in σ. For these schedules, there are two cases, and in each case, a comparison will be made between them.

First case: If $p_i \le p_j$, $d_i \le d_j$ produces that $s_i \le s_j$ for every $, j = 1,2, \dots , n$.

In this situation, there are the following: From the condition of the processing times, we ensure that:

$$\sum C_k(\sigma) \le \sum C_k(\dot{\sigma}) \qquad (1).$$

From the condition of slack time $s_i \le s_j$, there are $E_{max}(\sigma) \le E_{max}(\dot{\sigma})$ then $\sum E_k(\sigma) \le \sum E_k(\dot{\sigma})$.

From $p_i \le p_j$ and $d_i \le d_j$, this means $\sum V_k(\sigma) \le \sum V_k(\dot{\sigma})$, and the condition on the due date ensures that: $T_{max}(\sigma) \le T_{max}(\dot{\sigma})$ and $\sum V_k(\sigma) \le \sum V_k(\dot{\sigma})$ (2).

Hence, $\sum C_k(\sigma) + \sum V_k(\sigma) + \sum E_k(\sigma) + E_{max}(\sigma) + T_{max}(\sigma) \le \sum C_k(\dot{\sigma}) + \sum V_k(\dot{\sigma}) + \sum E_k(\dot{\sigma}) + E_{max}(\dot{\sigma}) + T_{max}(\dot{\sigma})$.

Second case: If $p_i \le p_j$ and $d_i \le d_j$, then it yields that $s_i > s_j$ for every $, j = 1,2, \dots , n$.

In this situation, there is the following: From the condition on the processing times, we ensure that (1) is satisfied, and the addition in cost is obtained from (1) is equal to $p_j - p_i$ which means $\sum C_k(\dot{\sigma}) = \sum C_k(\sigma) + p_j - p_i$ (3).

Then, $d_i - C_i(\sigma) = d_i - C_j(\dot{\sigma}) + p_j - p_i$, since $C_i(\sigma) = C_j(\dot{\sigma}) - p_j + p_i$, and $s_i = d_i - p_i \ge s_j = d_j - p_j$ then $d_i - p_i - C_j(\dot{\sigma}) \ge d_j - p_j - C_j(\dot{\sigma})$. Hence, $d_i - p_i + p_j - C_j(\dot{\sigma}) \ge d_j - C_j(\dot{\sigma})$, from this deduce that $E_{max}(\sigma) \ge E_{max}(\dot{\sigma})$ then $\sum E_k(\sigma) \ge \sum E_k(\dot{\sigma})$. Also, the obtained cost from this inequality is equal to $s_i - s_j$ which gives:

$E_{max}(\sigma) = E_{max}(\dot{\sigma}) + (s_i - s_j)$ and $\sum E_k(\sigma) = \sum E_k(\dot{\sigma}) + (s_i - s_j)$ (4).

.

Since $p_i \le p_j$ then $p_j - p_i \ge 0 \quad \forall i, j$ (5).

Since $d_i \le d_j$ then $d_j - d_i \ge 0 \quad \forall i, j$

From $s_i - s_j \le p_j - p_i$, then $E_{max}(\dot{\sigma}) + (s_i - s_j) \le E_{max}(\dot{\sigma}) + p_j - p_i$ and $\sum E_k(\dot{\sigma}) + (s_i - s_j) \le \sum E_k(\dot{\sigma}) + p_j - p_i$ (6).

(by adding for both sides) By

We add $\sum E_k(\dot{\sigma})$to both sides of (4), then we get $\sum C_k(\sigma) + \sum E_k(\sigma) + E_{max}(\sigma) \le \sum C_k(\sigma) + p_j - p_i + \sum E_k(\dot{\sigma}) + E_{max}(\dot{\sigma})$.

We add $\sum C_k(\sigma)$ to both sides. From (2), we get

$$\sum C_k(\sigma) + \sum E_k(\sigma) + E_{max}(\sigma) \le \sum C_k(\dot{\sigma}) + \sum E_k(\dot{\sigma}) + E_{max}(\dot{\sigma}) \qquad (7).$$

by adding T_{max} for both sides and from $p_i \le p_j$ and $d_i \le d_j$, this means $\sum V_k(\sigma) \le \sum V_k(\dot{\sigma})$, from (2) hence $\sum C_k(\sigma) + \sum E_k(\sigma) + E_{max}(\sigma) + T_{max}(\sigma) \le \sum C_k(\dot{\sigma}) + \sum E_k(\dot{\sigma}) + E_{max}(\dot{\sigma}) + T_{max}(\dot{\sigma})$.

Example (2): We 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	13	4	3	8	12	3
d_j	13	23	23	27	29	13
s_j	0	19	20	19	17	10

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

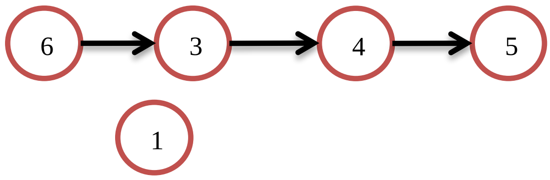

Figure 1: Dominant rule is shown in Example(2).

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

$$A = \begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & 0 & a_{23} & 1 & 1 & 0 \\ a_{31} & 1 & 0 & 1 & 1 & 0 \\ a_{41} & 0 & 0 & 0 & 1 & 0 \\ a_{51} & a_{52} & 0 & 1 & 0 & 0 \\ a_{61} & 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \text{ where } a_{ji} = \begin{cases} 1, \text{if } a_{ij} = 0 \\ 0, \text{if } a_{ij} = 1 \end{cases} .$$

Table 2: The potential efficient sequences are subject to the dominant rule in Example (2).

	EF.SE. W. DR						$(S_{(CVE)}M_{ET})$	(SP)
Seq	1	2	3	4	5	6	$(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max})$	$\sum C_j + \sum V_j + \sum E_j + E_{max} + T_{max}$
1	1	6	3	2	4	5	(146,26,6,3,14)	195
2=EDD	6	1	3	2	4	5	(135,19,14,10,14)	192
3	6	3	1	2	4	5	(125,22,27,17,14)	205
4	6	3	2	1	4	5	(116,26,40,17,14)	213
5	6	3	2	4	1	5	(111,25,55,17,18)	226
6=SPT	6	3	2	4	5	1	(110,14,49,17,30)	226

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

The sequences (1- 6) provide the problem $(S_{(CVE)}M_{ET})$ an efficient value that can be shown in Table 2, observe that the sequence number (2) in Table 2 provides an optimal value for the problem (SP).

5. 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 short processing time rule gives an efficient solution for the problem $(S_{(CVE)}M_{ET})$, and the optimal solution for the problem (sub-problem), this is proved in Proposition (1).
- Every optimal solution for the problem (sub-problem) is an efficient solution to the problem $(S_{(CVE)}M_{ET})$. This is proved in Proposition (2).
- The short processing time schedule σ gives an efficient solution for problem $(S_{(CVE)}M_{ET})$ and an optimal solution for problem(sub-problem) when one of the following conditions

is fulfilled: 1) $p_j = p, d_j = jp, \forall j, (j = 2,3, \dots, n)$. 2) $p_1 = d_1$ and $p_j = d_j - d_{j-1}, \forall j, (j = 2,3, \dots, n)$ 3) $C_j \le d_{\rho_j}, \rho = SPT = MST, \forall j$.

- Any schedule α gives an efficient solution for problem $(S_{(CVE)}M_{ET})$ and optimal solution for problem (sub-problem) when $p_j = p, d_j = d, and\ d \ge p$ for all j in schedule .

6. Conclusions and Future Works

In this study, a mathematical model was created to address the research problems *1//F*$(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max})$, *1//* $\sum C_j + \sum V_j + \sum E_j + E_{max} + T_{max}$. It has been proven that certain rules provide efficient (optimal) solutions to the $(S_{(CVE)}M_{ET})$ and (sub-problem) problems, finding and proving certain cases that discover some efficient (optimal) solutions for $(S_{(CVE)}M_{ET})$ and (sub-problem) the problem under consideration and demonstrating that the short processing time and give earliest due date efficient (optimal) solutions to these problems, demonstrated the significance of the Dominance Rule that can be used in this problem to improve efficient solutions, and Suggest some problems to be discussed and analyzed in future work:

1) *1/*S_f*/F*$(\sum C_j, \sum V_j, \sum E_j, E_{max}, T_{max})$.
2) *1/*S_f*/* $\sum C_j + \sum V_j + \sum E_j + E_{max} + T_{max}$.

References

[1] P. J.-C. B. Associate Professor Vincent T'kindt, *Multicriteria Scheduling Theory, Models and Algorithms Translated*, Second Edi. Springer Berlin Heidelberg New York.

[2] H. C. Joksch, "Constraints , Objectives , Efficient Solutions And Suboptimization In Mathematical Programming Author (s): H . C . JOKSCH Source : Zeitschrift für die gesamte Staatswissenschaft / Journal of Institutional and Published by : Mohr Siebeck GmbH & Co . KG S," *J. Institutional Theor. Econ.*, no. Januar 1966, pp. 5–13, 2016, [Online]. Available: http://www.jstor.org/stable/40748933

[3] 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, doi: 10.30526/34.3.2677.

[4] J. A. Hoogeveen, "Minimizing maximum promptness and maximum lateness on a single machine," *Math. Oper. Res.*, vol. 21, no. 1, pp. 100–114, 1996, doi: 10.1287/moor.21.1.100.

[5] W. E. Smith, "Various optimizers for single-stage production," *Navel Res. Logist. Q.*, vol. 3, pp. 59–66, 1956, doi: 10.1002/nav.3800030106.

[6] 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, doi: 10.24996/ijs.2020.61.8.21.

[7] 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, doi: 10.1287/opre.40.3.586.

[8] 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, doi: ISSN: 1609 - 4042 - Electronic ISSN: 2521-340.

[9] Z. M. Ali and T. S. Abdul Razaq, "Minimizing The Total Completion Times, The Total Tardinessand The Maximum Tardiness," *Ibn Al-Haitham J. Pure Appl. Sci.*, vol. 28, no. 2, pp. 155–170, 2015, doi: 10.30526/34.3.2677.

[10] M. Hashim, F. Hassan, and H. Ali, "Solving multi-objectives function problem using branch and bound and local search methods," *Int. J. Nonlinear Anal. Appl.*, vol. 13, no. 1, pp. 1649–1658, 2022, [Online]. Available: http://dx.doi.org/10.22075/ijnaa.2022.5780%0ASolving

[11] M. G. Ahmed and F. H. Ali, "Exact Method with Dominance Rules for Solving Scheduling on a Single Machine Problem with Multiobjective Function," *Al-Mustansiriyah J. Sci.*, vol. 33, no. 2, pp. 56–63, 2022, doi: 10.23851/mjs.v33i2.1091.

[12] 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*, pp. 103–108, doi: 10.1109/IEC49899.2020.9122821.

[13] 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, doi: 10.11591/ijeecs.v26.i1.pp334-341.

[14] T. S. Abdul-Razaq and A. O. Akram, "Local Search Algorithms for Multi-criteria Single Machine Scheduling Problem," *Ibn AL-Haitham J. Pure Appl. Sci.*, pp. 436–451, 2018, doi: 10.30526/2017.ihsciconf.1817.

[15] 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, doi: 10.24996/ijs.2020.SI.1.17.

[16] B. Atiya, A. J. K. Bakheet, I. T. Abbas, M. R. A. Bakar, L. L. Soon, and M. Bin Monsi, "Application of simulated annealing to solve multi-objectives for aggregate production planning," *AIP Conf. Proc.*, vol. 1739, no. November, 2016, doi: 10.1063/1.4952566.

[17] G. Strang, *Introduction to Linear Algebra, Fourth Edition*. 2009. [Online]. Available: http://students.aiu.edu/submissions/profiles/resources/onlineBook/Y5B7M4_Introduction_to_Linear_Algebra-_Fourth_Edition.pdf