Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

Solving tri-criteria: total completion time, total late work, and maximum earliness by using exact, and heuristic methods on single machine scheduling problem

Nagham M. Neamah [1,2*], Bayda A. 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

* Corresponding Author: Nagham M. Neamah

DOI: https:// doi.org/10.52866/ijcsm.2024.05.03.002
Received January 2024; Accepted March 2024; Available online June 2024

Abstract: The presented study investigated the scheduling regarding n jobs on a single machine. Each n job will be processed with no interruptions and becomes available for the processing at time 0. The aim is finding a processing order with regard to jobs, minimizing total completion time $\sum C_j$, total late work $\sum V_j$, and maximal tardiness E_{max} which is an NP-hard problem. In the theoretical part of the present work, the mathematical formula for the examined problem will be presented, and a sub-problem of the original problem of minimizing the multi-objective functions $\sum C_j + \sum V_j + E_{max}$ is introduced. Also, then the importance regarding the dominance rule (DR) that could be applied to the problem to improve good solutions will be shown. While in the practical part, two exact methods are important; a Branch and Bound algorithm (BAB) and a complete enumeration (CEM) method are applied to solve the three proposed MSP criteria by finding a set of efficient solutions. The experimental results showed that CEM can solve problems for up to $\mathrm{n} = 11$ jobs. Two approaches of the BAB method were applied: the first approach was BAB without dominance rule (DR), and the BAB method used dominance rules to reduce the number of sequences that need to be considered. Also, this method can solve problems for up to $n = 20$, and the second approach BAB with dominance rule (DR), can solve problems for up to $n = 60$ jobs in a reasonable time to find efficient solutions to this problem. In addition, to find good approximate solutions, two heuristic methods for solving the problem are proposed, the first heuristic method can solve up to $n = 5000$ jobs, while the second heuristic method can solve up to $n = 4000$ jobs. Practical experiments prove the good performance regarding the two suggested approaches for the original problem. While for a sub-problem the experimental results showed that CEM can solve problems for up to $n = 10$ jobs, the BAB without dominance rule (DR) can solve problems for up to $n = 15$, and the second approach BAB with dominance rule (DR), can solve problems for up to $n = 30$ jobs in a reasonable time to find efficient solutions to this problem. Finally, the heuristic method can solve up to $n = 4000$ jobs. Arithmetic results are calculated by coding (programming) algorithms using (MATLAB 2019a)

Keywords: Multi-Criteria (MC), Multi-Objective Function (MOF), Exact Methods (EM), Heuristic Methods (HM).

1. INTRODUCTION

Scheduling, generally speaking, means assigning machines to jobs in order to complete all jobs under the imposed constraints[1] ., . The problem is to find the optimal processing order of these jobs on each machine to minimize the given objective function. The scheduling problem is a set of n jobs on a single machine. Even job $j, j \in N$, where $N = \{1, \dots, n\}$ has an integer processed time p_j, due date d_j. Given schedule $\delta = (\delta(1), \delta(2), \dots, \delta(n))$, then for each job j calculate the completion time by $C_1 = p_1$ and $C_j = \sum_{k=1}^{n} p_{\delta_k}$ for $j = 2,3, \dots, n$. The lateness time of the job j is defined by $L_j = C_j - d_{\delta_j}$. The earliness of job j is specified via $E_j = max\{0, -L_j\} = max\left\{d_{\delta_j} - C_j, 0\right\}$, tardiness for job j, $T_j = max\{0, L_j\}$, slack time for job j is defined by $s_j = d_j - p_j$, and late work through $V_j = min\left\{T_j, p_{\delta_j}\right\} = min\left\{C_j - d_{\delta_j}, p_{\delta_j}\right\}$. Thus, there is a total completion time $\sum_{j \in N} C_j$, total late work $\sum_{j \in N} V_j$ and maximal earliness $E_{max} = max_{j \in N}\{E_{max}\}$. With regard to the total completion time of $1//\sum C_j$ problem is minimized via SPT (short processing time) rule is optimum to Smith 1956 [2]. The maximal earliness regarding the $1//\sum E_j$ problem has been minimized via

 *Corresponding author: author@organization.edu.co

the MST (i.e. minimum slack time) rule[3]. With regard to maximal tardiness for $1//T_{max}$ problem is minimized via the earliest due date rule (EDD) to Jackson 1955[4][5], the problems $1//\sum E_j$, $1//\sum V_j$, and $1//\sum T_j$ are NP-hard. Any problem with cost functions as a sub-problem is NP-hard.

In machine scheduling optimization problems with three criteria based on competing for objective functions, the Efficient Pareto optimal solutions set is formed as a vicarious of one optimal solution[6]. This set contains one (or more) solution(s) that, according to the objective functions, are superior to any other solution(s) [7] . In the literature, there are two approaches for tri-criteria scheduling problems; the simultaneous approach and the hierarchical approach [8].The most important criteria in scheduling problems are total completion time, total late work, and maximum earliness. These criteria were considered in different conditions for jobs and machines. Since most scheduling problems are NP-hard [21], it makes sense to use heuristics to find an approximate solution to the problem. Smith [1] solved scheduling problem of bi-criteria. Tariq and Firas [9] suggested algorithms, which present all efficient solutions within effective range for problems $1//F(\sum f_i, f_{max})$, $1//F(\sum f_i, f_{max}, g_{max})$, in which $\sum f_i = \sum C_i$ and $f_{max}, g_{max} \in \{L_{max}, T_{max}, V_{max}\}$. Oyetunji and Oluleye [10] used a heuristic approach to minimize the total completion time and number of tardy jobs simultaneously on a single machine with release date. Doha and Tariq [11] used branch and bound method to minimize the total completion time and maximum late work, and maximum earliness $(\sum C_j + V_{max} + E_{max})$. Hasson and Yousif [12] used CEM and BAB methods to minimize the $(\sum C_j, \sum T_j, \sum V_j)$,and $(\sum C_j, \sum T_j, V_{max})$. Aseel et al., [13] branch and bound method (BAB) and heuristic Methods to minimize total completion time, total earliness $(\sum C_j, \sum E_j)$ in single-machine scheduling problems. Arik [14] introduced earliness/tardiness with grey processing times and common due date. Chachan and Aameed [15] used a BAB method and Local search algorithms to minimize the sum total completion time, total tardiness, total earliness and total late work $\sum(C_j + T_j + E_j + V_j)$. Ibrahim et al., [16] used a BAB method to minimize multi-objective function $\sum(E_j + T_j + C_j + U_j + V_j)$. Hassan et al., [17] used a heuristic algorithm to minimize the sum of total completion time, maximum earliness, and maximum tardiness in a single-machine scheduling. Kramer and Submarian [18] introduced a unified heuristic for a large class of earliness-tardiness scheduling problems.

In this paper presents three criteria total completion time, total late work, and maximal tardiness for scheduling problems. We started by organizing it as a tri-criteria mathematical model and proposed a sub-problem with three objectives from the original problem. then, proposed exact methods to solve these problems, and BAB without DR and BAB with DR were implemented. Furthermore, two heuristic methods have been proposed, which have been adopted to find efficient solutions for this problem in a reasonable time.

The remainder of this paper is outlined as follows: Section 2 describes the mathematical formulations of tri-criteria and analysis of the sub-problem for the proposed problem. In Section 3, the exact, approximate methods and algorithms for solving the two problems given in the previous section were presented. Section 4 validates the proposed model and demonstrates the effectiveness of the proposed strategy through computational study and results, presenting the Results and Discussion in Section 5,6. Finally, conclusions and lists of future works are provided in Section 7.

2.Mathematical Model

In this section, the mathematical model of the single-machine scheduling problem for tri-criteria and tri-objective functions is proposed. First, we introduced some basic symbols and rules used in this work paper.

ACT/S: Average of CPU-Time per second.
ANEFS: Average number of efficient solutions.
Av: Average.
BAB (WDR): Branch and bound method with dominance rules.
BAB (WODR): Branch and bound method without dominance rules.
CT/S: CPU-Time per second.
Earliest due date (EDD): Jobs are sequenced in non-decreasing order regarding their due dates d_j(where $d_1 \leq d_2 \leq \cdots \leq d_n$), this rule utilized for minimizing T_{max} for problem $1//T_{max}$ [19] [4] [5].
EFSO (Efficient solution)[13]: A schedule α^* is known as efficient solution or Pareto optimal or (non-dominated) If cannot find another schedule α satisfying $h_j(\alpha) \leq h_j(\alpha^*), j = 1,2,..,n$, With at least one of the above considered a strict disparity. Another way is that α^* is dominated by α [17].
EXN: Example number.
F_{CVE}: Objective function of $(T_{CV}M_E)$ problem, and F_{SP} is objective function of (SP) problem.
Feasible Schedule (FS): Any schedule $\beta \in S$ (where S is the set of all schedules) can be considered feasible if it satisfies the constraints of the problem.
Minimum Slack Time (MST): Jobs are sequenced in a non-decreasing order regarding their slack time $s_j = d_j - p_j$ (where $s_1 \leq s_2 \leq \cdots \leq s_n$). For minimizing E_{max} with the use of this rule [20].
MOF: Multi-objective function, and MCF is multi- criteria function.
NEFS: Number of efficient Solutions.

n_i: The number of jobs, where i is the number of problems tested.
OF: Objective function regarding MSP could be either maximized or minimized under all possible constraints.
Optimal (OP): The σ^* schedule is considered as optimal in the case when there isn't other schedule σ that satisfies $f_j(\sigma) \leq f_j(\sigma^*), j = 1, \ldots, k$ (k: No. of criteria), assuming a strict inequality for a minimum of one of the conditions that have been mentioned earlier. If not, then σ can be considered as dominant over σ^* [13].
OPV: Optimal value of problem (SP), and EFV**:** efficient value of problem$(T_{CV}M_E)$.
RL: 0 < Real < 1.
Shortest processing time (SPT): Jobs are sequencing in a non-decreasing order of processing times p_j $(\text{i.e.}\, p_1 \leq p_2 \leq \cdots \leq p_n)$, this rule has been well known for minimizing $\sum C_j$ for problem 1// $\sum C_j$ [2].

2.1 The Mathematical Model for the 1// $(\sum C_j, \sum V_j, E_{max})$ Problem.

The aim of the problem is finding an efficient solution that gives the minimum value of the tri-criteria total completion time $\sum C_j$, total late work $\sum V_j$, and maximal tardiness E_{max}, this problem is denoted by [21]:

$$\left.\begin{array}{l} F_{CVE} = Min\left(\sum C_j, \sum V_j, E_{max}\right) \\ \quad \text{subject to} \\ \left.\begin{array}{l} C_j \geq p_j(\beta), \\ C_j = \displaystyle\sum_{k=1}^{j-1} p_k(\beta) + p_j(\beta), \\ T_j \geq C_j - d_j(\beta), \\ E_j \geq d_j(\beta) - C_j, \\ V_j = min\{T_j, p_j(\beta)\}, \\ V_j \geq 0, E_j \geq 0, \text{and } T_j \geq 0, \end{array}\right\} j \text{ from 1 to } n \end{array}\right\} \qquad (1)$$

This problem is referred to the $(T_{CV}M_E)$-problem and it is complicated for determining an efficient solution because $\sum V_j$ was an NP-hard problem [22].

For $(T_{CV}M_E)$-problem, sub-problem can be concluded that 1//$(\sum C_j + \sum V_j + E_{max})$ problem that referred to the (SP)-problem, and it can be defined as follows*:*

$$\left.\begin{array}{l} F_{SP} = Min\left(\sum C_j + \sum V_j + E_{max}\right) \\ \quad \text{subject to} \\ \left.\begin{array}{l} C_1 = p_1(\beta), \\ C_j = \sum_{k=1}^{j-1} p_k(\beta) + p_j(\beta), \\ T_j \geq C_j - d_j(\beta), \\ E_j \geq d_j(\beta) - C_j, \\ V_j = min\{T_j, p_j(\beta)\}, \\ V_j \geq 0, E_j \geq 0, \text{and } T_j \geq 0, \end{array}\right\} j \text{ from 1 to } n \end{array}\right\} \qquad (2).$$

In the following proposition, show that every optimal solution for (SP)-problem is an efficient solution to the $(T_{CV}M_E)$-problem.

Proposition [21]**:** Every optimal solution of (SP)-problem is an efficient solution to the $(T_{CV}M_E)$-problem.

3.Methodology

This section is devoted to examine the approaches for solving the $(T_{CV}M_E)$-problem and (SP)-problem. For the exact approaches, the BAB is utilized as the main approach for solving the problems. Moreover, BAB without DR and BAB with DR were performed. Also, two heuristic methods were proposed and were adopted to find efficient solutions to this problem in a reasonable time.

3.1 Exact Solution for $(T_{CV}M_E)$-Problem and (SP)-Problem

We have presented two exact methods in this subsection. First method: The Complete enumeration method (CEM) was used as a simple approach that generates all of the feasible tables for chooses the optimal solution. Second method: The BAB method is the most popular scheduling solution approach; BAB is an illustration of the implicit enumeration method that could identify the optimal solution through methodically reviewing subsets of potential solutions. A search tree with nodes corresponding to these sub-sets has been utilized for describing BAB.

The BAB method includes the following basic procedures:

- ❖ The process of branching involves breaking the parent (original) problem down into at least two sub-problems. Sub-problems are expressed as nodes in a search tree. Alternatively, backward branching (in which the jobs are placed one after the other starting at the end).
- ❖ The process of determining a lower bound on a sub-problem's optimal solution (i.e., nodes) is known as bounding.
- ❖ The search strategy is a way of choosing a node in the search tree to branch from; typically, the branch is from the lower bound (LB) in the search tree, with the following:

■ If $LB \geq UB$, consequently, such subproblem cannot result in a better solution to the main problem. As a result, we do not need to create new branches from the branching tree's matching node.
■ If $LB < UB$, then reset UB to take the value LB, (i.e., replace UB with LB). This procedure is repeated until all nodes (sub-clusters) had been tested.

LB procedure: where N denotes the set of all jobs, S is the set of scheduled jobs, and U is the set un-scheduled jobs, after that this procedure is:

1- Start with empty set of scheduled jobs (i.e., $S = \emptyset$), and start sorting the jobs (one by one) until get $|S| = n - 1$, and the n^{th} the job will be added to the set S and after that solve the last sequence with the MST, at each one of the steps, calculate $\left(\sum C_j, \sum V_j, E_{max}\right)$
2- For the set U, jobs were sorted by SPT then calculated $\left(\sum C_j, \sum V_j, E_{max}\right)$ for all jobs in the sequence.

3.1.1 BAB Method to Solve the $(T_{CV}M_E)$-Problem

In this subsection, we will use two BAB techniques to solve this problem.

BAB without Dominance Rules (BAB(WODRs)) for the $(T_{CV}M_E)$-problem

This method could be summed up as follows: the lower bound LB for the non-sequenced section of each node will be based on the SPT rule, and the UB utilized will be based on the MST rule. The following are the BAB (WODR) steps:

Algorithm 1: BAB(WODRs) Algorithm

Step 1: Input n, p_j and d_j for $j = 1,2, \dots, n$.

Step 2: Let $\mathcal{S} = \varphi$, for any α define $F_{CVE}(\alpha) = \left(\sum C_j(\alpha), \sum V_j(\alpha), E_{max}(\alpha)\right)$.

Step 3: Calculate an upper bound UB of the problem $(T_{CV}M_E)$ through sorting jobs in $\alpha = MST$. Calculate $F_{CVE}(\alpha)$ for $j = 1,2, \dots, n$, let $UB_{CVE} = F_{CVE}(\alpha) = \left(\sum C_j(\alpha), \sum V_j(\alpha), E_{max}(\alpha)\right)$ at the parent node of the search tree.

Step 4: For each node of the search tree of BAB approach and every partial sequence σ of jobs, compute $LB(\sigma) =$ cost of sequenced jobs σ for objective function + cost of un-sequenced jobs that have been obtained by sequence jobs in the SPT rule (where $\sigma = SPT$).

Step 5: Branch from each node with $LB \leq UB$.

Step 6: At the last level of the search tree, get a set of solutions, if $F(\sigma)$ the result is indicated, σ is added to the set S unless they are dominated by efficient solutions that have been obtained previously in $\mathcal{S}$, this process is called S filtering S.

Step 7: End.

BAB (WODR) solved the $(T_{CV}M_E)$-problem up to $n = 20$ in a reasonable amount of time. Also, another BAB which is based on DR (BAB (WDR)) is introduced to reduce the number of open nodes which saves time and increases n number of solved problems, since the search tree size (i.e., the number of the nodes) grows as the number of (n) increases in the (BAB) approach, particularly in the branching scheme. Thus, it is necessary to decrease this size by removing irrelevant solutions or choosing intriguing ones. The problem is being when the complementary subset of the solutions is stored, one subset of the solutions is rejected. The goal of dominance rules is to reduce the available research on scheduling problems. Consequently, it shortens the search period and narrows the search area.

BAB with Dominance Rules (BAB(WDRs)) method for the $(T_{CV}M_E)$-problem

This method could be summarized as follows: each node's UB and LB for the un-sequenced portion will be based on the SPT rule. To decrease the number of branched (open) nodes, which saves time and increases the number of solved problems, this BAB depends on DR, by applying the following theory:

Theorem [21]: If $p_i \leq p_k$ and $d_i \leq d_k$ then there's an optimal schedule for (SP)-problem where the job i processing before the job k.

Algorithm 2: BAB(WDRs) Algorithm

Step 1: Input n, p_j and d_j for $j = 1,2, \dots, n$. Find adjacency matrix $\boldsymbol{A}$.

Step 2: Let $\mathcal{S} = \varphi$, for any α define $F_{CVE}(\alpha) = \left(\sum C_j(\alpha), \sum V_j(\alpha), E_{max}(\alpha)\right)$

Step 3: Calculate an upper bound UB of the problem $(T_{CV}M_E)$ by sorting jobs in $\alpha = SPT$. Calculate $F_{CVE}(\alpha)$ for $j = 1,2, \dots, n$, let $UB_{CVE} = F_{CVE}(\alpha) = \left(\sum C_j(\alpha), \sum V_j(\alpha), E_{max}(\alpha)\right)$ at the parent node of a search tree.

Step 4: For every node of the search tree of BAB approach and every one of the partial sequences $(\sigma = SPT)$ of jobs, compute $LB(\sigma) =$cost of sequenced jobs (σ) for objective functions + cost of un-sequenced jobs obtained by a sequence of jobs in SPT rule σ.

Step 5: Branch from every node with $LB \leq UB$ and check $i \rightarrow j$.

Step 6: At the last level of the search tree, get a set of solutions, if $F(\sigma)$ the result is indicated, σ are added to the set S unless they are dominated by efficient solutions that have been obtained previously in $\mathcal{S}$, this process is called S filtering $\mathcal{S}$.

Step 7: Stop.

3.1.2 BAB Method for the(SP)-Problem

For the (SP)-problem, use the same BAB that is used for the $(T_{CV}M_E)$-problem, with some modifications indicated by BAB. First, calculate UB for (SP)-problem s.t., $UB(\alpha = SPT) = F_{SP}(\alpha) = \sum C_j(\alpha) + \sum V_j(\alpha) + E_{max}(\alpha)$, then calculate the LB of any node consisting of sequence and un-sequence parts (obtained by SPT rule) s.t., $LB(\sigma = SPT) = F_{SP}(\sigma) = \sum C_j(\sigma) + \sum V_j(\sigma) + E_{max}(\sigma)$, where σ is the rule for un-sequenced jobs. Repeat these steps until an optimal solution is obtained from the root.

3.2 Heuristic Methods for $(T_{CV}M_E)$-Problem and (SP)-Problem

Many research academics use approximate or heuristic algorithms to handle scheduling problems fast and efficiently since almost all of them are NP-hard and solving them using a CEM or BAB technique could be time-consuming [23]. Any algorithm or strategy that searches for optimum (nearly optimum) solutions in a reasonable period without an optimality guarantee in many cases is referred to as a heuristic (or approximation) strategy. In this subsection, we proposed two heuristics methods for solving the $(T_{CV}M_E)$-problem and (SP)-problem that are discussed below:

3.2.1 $SE - T_{CV}M_E$ Heuristic method for Solving $(T_{CV}M_E)$-Problem and (SP)-Problem

$SE - T_{CV}M_E$ method was presented in this subsection to solve the $(T_{CV}M_E)$-Problem [24]. The objective function using the SPT rule was calculated firstly. Then put the third job in the second position and the other jobs are still ordered depending on the SPT rule and calculate the objective function, etc. until n sequences are obtained, then repeat the same procedures when using the EDD rule, as described below:

Algorithm 3: $SE - T_{CV}M_E$ Heuristic Algorithm

Step 1: Input: n, p_j *and* $d_j, j = 1,2, \dots, n, \mathcal{S} = \varphi$.

Step 2: Arrange the jobs in the SPT rule (β_1) and calculate $F_{11}(\beta_1) = \left(\sum C_j(\beta_1), \sum V_j(\beta_1), E_{max}(\beta_1)\right)$; $\mathcal{S} = \mathcal{S} \cup \{F_{11}(\beta_1)\}$.

Step 3: For $i = 2, \dots, n$, put the job i in first position of β_{i-1} to get β_i and calculate $F_{1i}(\beta_i) = \left(\sum C_j(\beta_i), \sum V_j(\beta_i), E_{max}(\beta_i)\right)$; $\alpha = \alpha \cup \{F_{1i}(\beta_i)\}$.

End;

Step 4: Arrange the jobs in the EDD rule (σ_1) and calculate $F_{21}(\sigma_1) = \left(\sum C_j(\sigma_1), \sum V_j(\sigma_1), E_{max}(\sigma_1)\right)$; $\mathcal{S} = \mathcal{S} \cup \{F_{21}(\sigma_1)\}$.

Step 5: For $i = 2, \dots, n$, put the job i in first position of σ_{i-1} to get σ_i and calculate $F_{2i}(\sigma_i) = \left(\sum C_j(\sigma_i), \sum V_j(\sigma_i), E_{max}(\sigma_i)\right)$; $\mathcal{S} = \mathcal{S} \cup \{F_{2i}(\sigma_i)\}$.
End;
Step 6: A filter set $\mathcal{S}$ to obtain a set of efficient solutions to problem $(T_{CV}M_E)$.
Step 7: Output: the set of efficient solutions $\mathcal{S}$.
Step 8: End.

3.2.2 $DR - T_{CV}M_E$ Method for Solving $(T_{CV}M_E)$-Problem and (SP)-Problem

The second heuristic method is $(DR - T_{CV}M_E)$ depends on dominance rules. $DR - T_{CV}M_E$ summarized by finding the sequence sort with a minimum of p_j and d_j, which isn't inconsistent with dominance rules, and calculating the objective function. $DR - T_{CV}M_E$ algorithms can be summarized in the following steps:

Algorithm 4: $DR - T_{CV}M_E$ Heuristic Algorithm

Step 1: Input: n, p_j and $d_j, j = 1,2, \dots, n$.
Step 2: Apply remark or theorem (1) to find the DRs and corresponding adjacent matrix A; $N = \{1,2, \dots, n\}$ calculate $s_j = d_j - p_j, \forall j \in N, \mathcal{S} = \varphi$.
Step 3: Find a sequence α_1 with a non-increasing order of p_j that does not conflict with DR (matrix A), if it is more than one job order α_1 by d_j, then $\mathcal{S} = \mathcal{S} \cup \{\alpha_1\}$.
Step 4: Find a sequence α_2 with a non-increasing order of d_j does not conflict with the DR (matrix A), if there is more than one job that breaks links arbitrarily order α_2 by p_j, then $\mathcal{S} = \mathcal{S} \cup \{\alpha_2\}$.
Step 5: Find the dominant sequence set $\mathcal{S}'$ from $\mathcal{S}$.
Step 6: Calculate $F_{CVE}(\mathcal{S}')$.
Step 7: Output: Effective solution set $\mathcal{S}'$.
Step 8: END

4. Practical Results of $(T_{CV}M_E)$-Problem and (SP)-Problem

In this section, the results for applying the Exact, heuristic methods for the $(T_{CV}M_E)$- problem and (SP)-problem will be compared. The CEM is tested by programming it using MATLAB 2019a. Since we deal with the MSP, so the p_j and d_j values are randomly generated for five examples s.t. $sp_j \in [1,10]$ and $d_j \in \begin{cases} [1,30] & 1 \le n \le 29 \\ [1,40] & 30 \le n \le 99 \\ [1,50] & 100 \le n \le 999 \\ [1,70] & \text{othierwise} \end{cases}$, with condition $d_j \ge p_j$, for $j = 1,2, \dots, n$.

4.1 Comparison Results of the $(T_{CV}M_E)$-Problem

This subsection introduced the results the exact and the heuristic methods for the $T_{CV}M_E$-problem. The efficient results of applying CEM method for the $(T_{CV}M_E)$-problem are presented in Table 1.

Table 1.- The efficient results of applying CEM for the $(T_{CV}M_E)$-problem when $n = 4,5, \dots, 10$.

EXN	CEM		
n_5	EF	TIME	NES
	$AV(F_{CVE})$	ACT/S	ANEFS
4	(57.4, 2.5,15.8)	RL	6.4
5	(88.2,7.2,12.3)	RL	7.2
6	(110.2,11.6,13.4)	RL	15.2
7	(128.1,14.2,10.9)	RL	25.6
8	(150.9,16.0,11.8)	RL	20.8
9	(216.4, 25.9,8.5)	8.2	21.2
10	(205.0,18.3, 12.1)	87.2	40
11	(301.0,35.5,8.3)	989.0	26.8

The efficient results of applying BAB without DR and BAB with DR for the $(T_{CV}M_E)$-problem, for different n are shown in Table 2.

Table 2.- A comparison results between BAB without DR and BAB with DR for $(T_{CV}M_E)$-problem, for different n.

EXN	BAB(WODR)LB=SPT, UB=MST			BAB(WDR)LB=SPT, UB=MST		
n_5	MOF	TIME	NES	MOF	TIME	NES
	$AV(F_{CVE})$	ACT/S	ANEFS	$AV(F_{CVE})$	ACT/S	ANEFS
4	(57.5,2.5,15.8)	RL	6.4	(57.5,2.5,16.5)	RL	5.4
5	(87.3,7.4,12.1)	RL	6.6	(99.7,12.7,13.9)	RL	3.2
6	(110.0,11.8,13.5)	RL	14.8	(115.6,18.3,14.5)	RL	9.6
7	(128.4,14.5,10.8)	RL	24.0	(127.2,18.2,11.1)	RL	11.0
8	(151.0,16.1,12.0)	RL	20.2	(125.7,14.4,13.9)	RL	13.0
9	(214.6,26.2,8.4)	RL	18.2	(190.0,27.5,11.1)	RL	6.8
10	(209.6,18.7,11.3)	RL	31.6	(192.3,19.3,13.3)	RL	19.2
11	(303.3,36.2,8.3)	RL	24.8	(280.7,36.5,9.7)	RL	12.4
12	(351.2,43.5,10.1)	RL	45.6	(332.5,48.2,11.9)	RL	20.8
13	(383.5,43.1,7.7)	RL	33.4	(352.7,44.6,11.4)	RL	17.0
14	(471.9,55.5,7.0)	1.0	32.6	(457.3,61.1,8.1)	RL	7.2
15	(509.3,57.3,9.2)	RL	45.0	(495.0,60.1,10.8)	RL	22.2
16	(538.6,60.5,9.0)	2.9	71.2	(521.6,67.2,13.5)	RL	18.2
17	(644.8,71.4,9.6)	264.7	69.6	(627.4,78.4,12.2)	RL	15.8
18	(711.5,74.4,11.1)	17.6	69.2	(692.4,80.4,13.3)	RL	28.0
19	(743.0,77.3,8.6)	36.5	48.6	(702.1,80.1,10.2)	RL	21.6
20	(825.5,78.8,10.8)	10541.0	96.8	(775.3,81.8,12.3)	RL	36.4
21	-	-	-	(828.7,83.6,12.0)	RL	31.4
22				(1045.1,106.6,11.7)	RL	20.0
23	-	-	-	(1227.2,114.3,10.3)	RL	18.4
24	-	-	-	(1270.8,119.7,14.1)	RL	34.6
25	-	-	-	(1419.2,129.1,11.0)	RL	21.4
30	-	-	-	(1937.2,154.7,10.4)	RL	17.2
40	-	-	-	(3318.9,206.8,8.2)	RL	29.4
50	-	-	-	(4950.5,255.8,11.4)	11.8	42.6
60	-	-	-	(6775.1,298.1,9.8)	488.4	31.4
70	-	-	-	(9714.7,363.3,11.3)	654.1	48.4

Comparison efficient results between $SE - T_{CV}M_E$ and $DR - T_{CV}M_E$ for the $(T_{CV}M_E)$-problem are shown in Table 3, for different n.

Table 3. Comparison efficient results between $SE - T_{CV}M_E$ and $DR - T_{CV}M_E$ for the $(T_{CV}M_E)$-problem,for different n.

EXN	$SE - T_{CV}M_E$			$DR - T_{CV}M_E$		
n_5	MCF	TIME	NES	MCF	TIME	NES
	$AV(F_{CVE})$	ACT/S	ANEFS	$AV(F_{CVE})$	ACT/S	ANEFS
4	(56.1,2.7,16.8)	RL	5.0	(58.8,4,15.6)	RL	3.6
5	(85.3,9.9,14.1)	RL	3.4	(88.1,8.4,12.4)	RL	3.4
6	(108.3,16.2,16.5)	RL	5.4	(114.4,13.9,13.5)	RL	4.2
7	(125.4,17.9,13.3)	RL	6.2	(135.5,15.8,12.4)	RL	5.2
8	(150.9,20.2,14.4)	RL	5.8	(161.7,17.9,12.9)	RL	5.8
9	(216.0,33.3,10.1)	RL	4.8	(247.6,30.6,9.8)	RL	7.4
10	(216.7,25.9,14.4)	RL	8.0	(234.1,22,12)	RL	6.2
11	(301.2,42.0,11.4)	RL	6.6	(347.9,39.7,9.8)	RL	9.2
40	(3784.3,213.2,11.1)	RL	17.2	(4278.1,206.6,9.1)	RL	16.6

60	(8112.9,311.9,9.1)	RL	18.4	(8944.9,303.1,9.9)	RL	18.0
80	(15712.3,439.7,10.9)	RL	24.6	(16660.2,428.5,10.0)	RL	20.4
100	(25252.1,560.8,10.3)	RL	29.4	(25593.4,548.9,8.9)	RL	17.6
400	(397689.0,2198.4,7.8)	1.3	49.8	(341777.7,2187.0,7.6)	1.6	20.4
600	(887871.4,3279.7,7.7)	2.5	53.2	(753053.7,3267.8,9.0)	3.5	34.0
800	(1585853.4,4415.1,7.1)	4.2	51.0	(1351124.3,4403.8,8.8)	6.6	40.4
1000	(2500484.0,5509.3,8.6)	6.3	67.6	(2095245.3,5498.9,8.5)	10.7	39.6
2000	(10096117.5,11005.7,9.6)	28.8	80.6	(8328351.8,10995,8.7)	78.3	38.0
3000	(22453017.4,16472.7,8.8)	83.2	73.0	(18748013.8,16462.3,8.7)	274.8	36.8
4000	(40284000,22045.4,9.2)	186.3	77.4	(33463280.3,22034.9,8.6)	658.1	37.8
5000	(62627693.8,27452.2,8.5)	344.9	72.2	-	-	-

4.2 Comparison results of (SP)-problem

In this subsection, the optimal results of applying the exact methods will be compared with the heuristic methods for the (SP)-problem.

The optimal results of applying BAB (WODR), and BAB (WDR) that were put to comparison with the CEM for the (SP)-problem have been listed in Table 4, for different n

Table 4.- Comparison optimal results of BAB (WDR) and BAB (DR) with CEM for (SP)-problem, for different n

EXN	CEM		BAB(WODR) LB=SPT=UB		BAB(WDR) LB=SPT=UB	
n_5	MOF	TIME	MOF	TIME	MOF	TIME
	$AV(F_{SP})$	ACT/S	$AV(F_{SP})$	ACT/S	$AV(F_{SP})$	ACT/S
4	72.2	RL	72.2	RL	72.2	RL
5	103.4	RL	103.4	RL	104.4	RL
6	128.8	RL	129.4	RL	133.6	RL
7	144.2	RL	144.2	RL	147.8	RL
8	169	1.0	169.0	RL	169.4	RL
9	242	9.1	242.0	RL	245.4	RL
10	224	95.7	224.4	RL	227.0	RL
11	329	1048.0	329.0	1.6	333.0	RL
12	-	-	388.8	10.2	393.8	RL
13	-	-	405.4	RL	408.2	RL
14	-	-	523.6	110.4	528.2	RL
15	-	-	529.0	599.9	554.4	RL
20	-	-	-	-	896.6	RL
21	-	-	-	-	1085.0	RL
22	-	-	-	-	1102.2	RL
23	-	-	-	-	1053.6	RL
24	-	-	-	-	1361.6	RL
25	-	-	-	-	1609.0	13.9
26	-	-	-	-	1460.2	4.2
27	-	-	-	-	1790.8	9.5
28	-	-	-	-	1648.8	2.9
29	-	-	-	-	2012.0	64.3
30	-	-	-	-	2110.4	599.8

The optimal results of applying $SE-SP$ and $DR-SP$ that were put to comparison with CEM for the (SP)-problem are presented in Table 5, n = 4, 5, …, 10.

Table 5.- Comparison optimal results between $SE - SP$ and $DR - SP$ with CEM for the (SP)-problem n = 4, 5, …, 10.

EXN	CEM		$SE - (SP)$		$DR - (SP)$	
n_5	MOF	TIME	MOF	TIME	MOF	TIME
	$AV(F_{SP})$	ACT/S	$AV(F_{SP})$	ACT/S	$AV(F_{SP})$	ACT/S
4	72.2	RL	72.4	RL	72.4	RL
5	103.4	RL	105.0	RL	106.6	RL
6	128.8	RL	135.4	RL	133.0	RL
7	144.2	RL	149.8	RL	151.6	RL
8	169	1.0	177.0	RL	178.4	RL
9	242	9.1	248.4	RL	252.2	RL
10	224	95.7	231.0	RL	230.4	RL

Table 6 presents the optimal results of applying $SE - SP$ and $DR - SP$ for the (SP)-problem, for different n.

Table 6.- Comparison optimal results between $SE - SP$ and $DR - SP$ for problem (SP) for different n.

EXN	$SE - (SP)$		$DR - (SP)$	
n_5	MOF	TIME	MOF	TIME
	$AV(F_{SP})$	ACT/S	$AV(F_{SP})$	$AV(F_{SP})$
40	3532.8	RL	3530.0	RL
60	7299.6	RL	7302.2	RL
80	13561.2	RL	13559.2	RL
100	21173.4	RL	21171.0	RL
400	312828.0	2.0	312818.0	1.9
600	697042.2	3.1	697034.8	4.3
800	1245294.0	5.6	1245287.2	9.5
1000	1940975.0	9.1	1940967.4	18.9
2000	7724281.6	14.8	7724272.6	30.4
3000	17330649.2	42.0	17330640.0	107.9
4000	30937071.0	91.4	30937060.4	257.5

5. Results and Discussion

Analyze the results by discussing the $(T_{CV}M_E)$-problem:

- The CEM starts to give the minimum values for the $1//\left(\sum C_j, \sum V_j, E_{max}\right)$ problem compared to the results for the BAB up to $n \leq 11$. Also, CEM takes a long time in the CPU-Time compared to BAB. Moreover, BAB (WODR) starts giving the minimum values for the $1//\left(\sum C_j, \sum V_j, E_{max}\right)$ problem compared to the results for BAB (WDR) , for $n \leq 7$, look at Tables 1, 2. while, BAB (WDR) performs better than BAB(WODR), for $n > 7$, and the BAB without DR solved the problem in all cases from $n = 4$ to $n = 20$, but failed to solve the problem when $n > 20$, while BAB with DR solved the problem in all cases from $n = 4$ to $n = 70$, but failed to solve the problem when $n > 70$.

- The CEM gives better results compared to the heuristic method $SE - T_{CV}M_E$, $DR - T_{CV}M_E$ for problem $1//\left(\sum C_j, \sum V_j, E_{max}\right)$. In addition, CEM takes a long time in the CPU-Time, while the heuristic method $SE - T_{CV}M_E$ gives better results than $DR - T_{CV}M_E$ for $n \leq 11$, look at Table 1,3.

- The heuristic method $SE - T_{CV}M_E$ gives better results than $DR - T_{CV}M_E$ for problem $1//\left(\sum C_j, \sum V_j, E_{max}\right)$ up to $n \leq 400$, while heuristic method $DR - T_{CV}M_E$ gives better results than $SE - T_{CV}M_E$ for problem $T_{CV}M_E$ up to $n \geq 400$, observe Table 3. Furthermore, $DR - T_{CV}M_E$ was successful in resolving all issues $n \leq 4000$ and failed to solve all problems for $n > 4000$, $SE - T_{CV}M_E$ was successful in resolving all issues $n \leq 5000$ and failed to solve all problems for $n > 5000$.

Analyze the results by discussing the (SP)-problem:

- The results of applying BAB (WODR) are identical to CEM, but CEM takes a long time compared to BAB (WODR) and starts to give minimum values for the problem $1//\sum C_j + \sum V_j + E_{max}$ compared to the results of BAB (WDR). In addition, BAB (WODR) takes a long time in CPU time compared to BAB (WDR), look at Tables 4. Also, CEM was successful in resolving all issues $n \leq 11$, and failed to solve all problems for $n > 11$, BAB (WODR) was successful in resolving all issues $n \leq 15$, and failed to solve all problems for $n > 15$, and BAB (WDR) was successful in resolving all issues $n \leq 30$, and failed to solve all problems for $n > 30$

- The CEM gives better results than the heuristic methods $DR - SP$, and $SE - SP$. Also, CEM takes a long time compared to $DR - SP$ and $SE - SP$ for problem SP, and $SE - SP$ gives better results than $DR - SP$ for problem $1//\sum C_j + \sum V_j + E_{max}$ up to $n = 4:10$. On the other hand, heuristic methods solve problems with $n \leq 4000$ and require less processing time, look at the Tables 4,6.

- In general, the BAB application results are better when compared to heuristic methods, more specifically the BAB (WODR) application results are better when compared to those of the BAB(WDR) application and heuristic methods, which also shows that BAB with DR solved all cases of the problem from n = 4 to n = 15, and BAB without DR failed to solve all problems for $n > 15$, see Tables 4,5,6.

- The heuristic method $DR - SP$ gives better results than $SE - SP$ for problem $1//\sum C_j + \sum V_j + E_{max}$ for different n, which also takes a long time compared to $SE - SP$, moreover. Heuristic strategies were successful in resolving all issues $n \leq 4000$ and failed to solve all problems for $n > 4000$, see Tables 6.

6. Main Results

1. The practical results of this research demonstrated the efficiency of the proposed methods: exact methods (CEM and BAB) for the two problems.
2. For $1//(\sum C_j, \sum V_j, E_{max})$ problem, $n \leq 11$, CEM performs better than BAB and heuristic methods in accuracy but takes a long time in the CPU-Time, for $n \leq 7$, the performs of BAB(WODR) is better than BAB(WDR), while, BAB (WDR) performs better than BAB(WODR), for $n > 7$. In addition, for $n \leq 400$, the performs of $SE - T_{CV}M_E$ is better than $DR - T_{CV}M_E$, while, $DR - T_{CV}M_E$ performs better than $SE - T_{CV}M_E$, for $n \geq 400$.
3. For $1//\sum C_j + \sum V_j + E_{max}$ problem, $n \leq 11$, the performs of CEM and BAB (WODR) are identical and, better than BAB (WDR) and heuristic methods in accuracy, but CEM takes a long time in the CPU-Time. In addition, the performs of $DR - T_{CV}M_E$ is better than $SE - T_{CV}M_E$, for all n.

7. Conclusions and Future Works

In this paper, the tri-criteria machine scheduling problems are solved, where the discussed problem is represented by the $1//(\sum C_j, \sum V_j, E_{max})$, and from this problem can be derived a sub-problem $1//(\sum C_j + \sum V_j + E_{max})$. In this paper, two techniques of BAB methods are introduced to solve the two problems $(T_{CV}M_E)$ and (SP) problems, with and without dominance rulers, and the results demonstrate the accuracy of the BAB results. Introduced two new heuristic methods $SE - T_{CV}M_E$ and $DR - T_{CV}M_E$ with good performance for the two studied problems.
The next MSPs will make intriguing subjects for future study.

- *1//* $Lex(\sum C_j, \sum V_j, E_{max})$.
- *1//* $Lex(\sum V_j, \sum C_j, E_{max})$.
- *1//* $Lex(E_{max}, \sum C_j, \sum V_j)$.

Funding
None

Acknowledgement
All thanks to Allah Almighty, who allowed me to successfully accomplish this effort.

Conflicts of Interest
The authors declare no conflict of interest.

References
[1] 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.

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

[3] 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, doi: https://doi.org/10.1006/jagm.1996.0051.

[4] Leslie A. Hall and David B. Shmoys, "JJackson's Rule for Single-Machine Scheduling: Making a Good Heuristic Better," *Math. Oper. Res.*, vol. 17, no. 1, pp. 22–35, 1992, [Online]. Available: http://www.jstor.org/stable/3689890 Accessed:

[5] B. J. Lageweg, J. K. Lenstra, and A. H. G. R. Kan, "Minimizing maximum lateness on one machine: computational experience and some applications," *Stat. Neerl.*, vol. 30, no. 1, pp. 25–41, 1976, doi: 10.1111/j.1467-9574.1976.tb00264.x.

[6] Kenneth R. Baker and Dan Trietsch, *Principles of Sequencing and Scheduling*, Second Edi. Set in 10/12pt Warnock by SPi Global, Pondicherry, India Printed: © 2019 John Wiley & Sons, Inc. Edition, 2009.

[7] 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.

[8] M. Cosor, T. U. Eindhoven, and D. Version, "Polynomial-time algorithms for single-machine bicriteria scheduling," vol. 9221, no. June, 1992.

[9] 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.

[10] E. O. Oyetunji and A. E. Oluleye, "Heuristics for minimizing total completion time on single machine with release time," *Adv. Mater. Res.*, vol. 18–19, no. June 2007, pp. 347–352, 2007, doi: 10.4028/www.scientific.net/amr.18-19.347.

[11] Tariq S.Abdul-Razaq and Doha A.Abbas, "A Function Of Two Or Three Cost Criteria To Be Optimized," *Al-Mustansiriyah J. Sci.*, vol. 24, no. 2, pp. 113–132, 2013.

[12] Saad Talib Hassn and Shahbaa Mohammed Yousif, "Minimizing Multiple Objective Function for Scheduling Machine Problems," *Math. Theory Model.*, vol. 5, no. 6, pp. 49–72, 2015.

[13] 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.

[14] O. A. Arık, "Single machine earliness/tardiness scheduling problem with grey processing times and the grey common due date," *Grey Syst.*, vol. 11, no. 1, pp. 95–109, 2021, doi: 10.1108/GS-01-2020-0010.

[15] A. S. Hameed and Hanan Ali Chachan, "Genetic Algorithm and Particle Swarm Optimization Techniques for Solving Multi-Objectives on Single Machine Scheduling Problem," *Ibn AL- Haitham J. Pure Appl. Sci.*, vol. 33, no. 1, p. 119, 2020, doi: 10.30526/33.1.2378.

[16] 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

[17] 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.

[18] A. Kramer and A. Subramanian, *A unified heuristic and an annotated bibliography for a large class of earliness–tardiness scheduling problems*, vol. 22, no. 1. 2019. doi: 10.1007/s10951-017-0549-6.

[19] A. O. A. Tariq S.Abdul-Razaq, "Local Search Algorithms for Multi-criteria Single Machine Scheduling Problem," *Ibn AL-Haitham J. Pure Appl. Sci.*, vol. IHSCICONF, no. Special Issue, pp. 436–451, 2017, doi: 10.30526/2017.ihsciconf.1817.

[20] 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.

[21] N. M. Neamah and B. A. Kalaf, "Solving the multi-criteria: total completion time, total late work, and maximum earliness problem," *Period. Eng. Nat. Sci.*, vol. 11, no. 3, pp. 46–57, 2023, doi: 10.21533/pen.v11i3.3559.g1288.

[22] C. N. Potts and L. N. Van Wassenhove, "Approximation algorithms for scheduling a single machine to minimize total late work," *Oper. Res. Lett.*, vol. 11, no. 5, pp. 261–266, 1992, doi: 10.1016/0167-6377(92)90001-J.

[23] F. H. Ali and 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 2020*, no. 1, pp. 152–156, 2020, doi: 10.1109/CSASE48920.2020.9142053.

[24] M. Ahmed and F. Ali, "Efficient algorithms to solve tricriteria machine scheduling problem," *J. Al Rafidain Univ. Coll. (15th 2nd Int. Conf. Stat. Appl.*, vol. 46, pp. 485–493, 2020.