International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 11, No. 4, April, 2024, pp. 987-995
Journal homepage: http://iieta.org/journals/mmep

Solving Tri-criteria: Total Completion Time, Total Earliness, and Maximum Tardiness Using Exact and Heuristic Methods on Single-Machine Scheduling Problems

Nagham M. Neamah[1,2*], Bayda A. Kalaf[2], Wafaa Mansoor[3]

[1] Mathematics Department, College of Science for Women, University of Baghdad, Baghdad 10071, Iraq
[2] Mathematics Department, College of Education for Pure Science Ibn-Al-Haitham, University of Baghdad, Baghdad 10071, Iraq
[3] Mathematics and Statistics, Murdoch University, Perth 6150, Australia

Corresponding Author Email: naghamm_math@csw.uobaghdad.edu.iq

https://doi.org/10.18280/mmep.110415

Received: 20 November 2023
Revised: 15 January 2024
Accepted: 30 January 2024
Available online: 26 April 2024

Keywords:
multi-criteria (*MC*), *multi-objective function* (*MOF*), *exact methods* (*EMs*), *heuristic methods* (*HMs*)

ABSTRACT

Machine scheduling problems have become increasingly complex and dynamic. In industrial contexts, managers often evaluate several objectives simultaneously and attempt to identify the optimal solution that satisfies all concerns. This study proposes two heuristic methods based on SPT and dominated rules (DR) to minimize Total Completion $\sum C_j$, Total Earliness $\sum E_j$, and Maximum Tardiness Time T_{max} for multi-criteria and multi-objective functions $(1//(\sum C_j, \sum E_j, T_{max})$ and $(\sum C_j + \sum E_j + T_{max}))$ based on single machine scheduling problems. in addition, two exact methods Branch and Bound (BAB with and without DR) and a complete enumeration method are applied to solve the multi- criteria and multi-objective functions. According to the calculation results, the CEM is able to solve problems up to $n = 11$ jobs, while BAB without DR and BAB with DR able to resolve problems from $n = 19$ to $n = 50$ jobs, respectively, within a reasonable time. However, heuristic methods can solve up to $n = 5000$ jobs. in addition, the experimental results for a subproblem show that the heuristic methods can solve up to $n = 4000$ jobs. Practical experiments demonstrate the proposed heuristic methods are the most effective of all approaches. All methods used in this work were coded with MATLAB 2019a.

1. INTRODUCTION

Since 1954, scheduling issues have received a great deal of attention in the literature. Assigning machines to jobs in order to finish them all within specified constraints is the general definition of scheduling [1, 2]. Efficient scheduling is essential to prevent excessive or underutilization of resources [3]. The scheduling problem is a collection of n jobs performed by one machine. Given job $j, j \in N$, where $N = \{1, \dots, n\}$ has an integer-processed time p_j, due date d_j. Given schedule $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(n))$, $C_1 = p_1$ and $C_j = \sum_{k=1}^{n} p_{\sigma_k}$ (where $j = 2, 3, \dots, n$) are then used to determine the completion time for each job j. $L_j = C_j - d_{\sigma_j}$ expresses the lateness time of the job j. $E_j = max\{0, -L_j\} = max\{d_{\sigma_j} - C_j, 0\}$ is used to defined the earliness of job j, tardiness in job j is defined as $T_j = max\{0, L_j\}$, $s_j = d_j - p_j$ is used to determine the slack time of job j. Thus, there is a total completion time $\sum_{j \in N} C_j$, total earliness time $\sum_{j \in N} E_j$ and maximal tardiness $T_{max} = max_{j \in N}\{T_{max}\}$. Smith [4] concerned the total completion time, $1// \sum C_j$ problem is minimized using SPT (short processing time) rule and is optimum in 1956. The maximal earliness regarding the $1//\sum E_j$ problem has been minimized using MST (minimum slack time) rule [5]. According to a study by Jackson [6], the Earliest Due Date (EDD) rule was used to minimize the maximum tardiness with respect to the $1//T_{max}$; the problem $1//\sum E_j$ is NP-hard. Any problem with cost functions as subproblems is NP-hard. The challenge is to determine the ideal processing sequence for these jobs on each machine in order to minimize the given objective function. Researchers focused on only one objective function [7]. In practical cases, the decision maker only needs to select one objective function. Nowadays, more studies are conducted on multi-objective planning problems. An overview of multiple and binary scheduling problems was published by Hoogeveen [8]. Hierarchical and simultaneous minification are the two main structures used to solve competing criteria [9]. The primary criterion is the first and the secondary criterion is the second. In this scenario, one decreases the primary criterion and selects a table with the minimum value of the secondary criterion. In the second method, a Pareto set is formed and the decision maker is the one with the optimal composite objective function [10]. Hoogveen [8] presented an algorithm that finds all effective tables for $1//(\sum C_j, F_{max})$ problem. Abdul-Razaq and Ali [11] studied problem $1//(\sum C_j, \sum T_j, T_{max})$ and found a sub-problem, also solved this problem by branch and bound method. Abdul-Razaq and Motair [9] presented a multi-objective function $1// \sum C_j + \sum T_j + T_{max} + E_{max}$ and used branch and bound to minimize this problem. Jawad et al. [12] provided the BAB to solve multi-criteria objective function $1//\left(\sum C_j, \sum E_j\right)$ problem in the SMSP. Ahmed and Ali [13] suggested BAB and heuristic approaches to minimize the $\sum C_j + R_L + T_{max}$ for single machine scheduling problem. Al-Tameemi [14] used BAB to solved the problem $1//\sum C_j + \sum T_j + E_{max}$. Arik [15] offered earliness /tardiness with shared due dates and gray processing times. Hameed and Chachan [16] multi-objective minimization the $\sum\left(C_j + T_j + E_j + V_j\right)$ was proposed for single machine scheduling problems, two local search algorithms (GA and PSO) were also used. Also, Chachan and Hameed [17] used BAB method and local search algorithms to minimize $\sum\left(C_j + T_j + E_j + V_j\right)$. In addition, Chachan and Jaafer [18] presented a Branch and Bound algorithm to minimize the $\sum\left(E_j + T_j + C_j + U_j + V_j\right)$ with unequal release dates for scheduling (n) jobs on a single machine. Large and complex problems in the research community are often solved using contemporary heuristic optimization techniques [19]. Hassan et al. [20] used a heuristic algorithm to minimize the $\left(E_{max} + T_{max} + \sum C_j\right)$in a SMSP. Neamah and Kalaf [21] proved that SPT and EDD rules give efficient (optimal) solutions for two problems $1//\left(\sum C_j, \sum V_j, E_{max}\right)$, and $1//\sum C_j + \sum V_j + E_{max}$, also, proven special cases, resulting in the most an efficient and optimal solution to these problems.

Within the paper, proposed two new heuristic methods to solve and find efficient solutions three criteria $\sum C_j, \sum V_j, E_{max}$ 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.

Below is an outline of the remaining portion of this paper: Section 2 describes the mathematical formulations of tri-criteria and analysis of the sub-problem for the proposed problem. Section 3 presents the exact, approximate methods and algorithms for solving the two problems given in Section 2. Section 4 validates the proposed model and demonstrates the effectiveness of the proposed strategy through computational study and results. Moreover, Section 4 presents the results and accompanying discussion. Finally, conclusion and lists of future works are provided in Section 5.

2. MATHEMATICAL MODEL

The mathematical formulation of the single machine scheduling problem for tri-criteria and tri-objective functions is presented in this section. Firstly, some of the notations included that are utilized in the formulation of tri-criteria and tri-objective functions of the single machine scheduling problem:

ACT/S: Average of CPU-Time per second.

ANEFS: Average number of efficient solutions.

BAB(WDR): BAB method with dominance rules (DRs).

BAB(WODR): BAB method without DRs.

CT/S: CPU-Time per second.

EDD: Jobs are arranged according to their due dates in non-descending order d_j (where $d_1 \leq d_2 \leq \cdots \leq d_n$); this rule is utilized for minimizing T_{max} for problem $1//T_{max}$ [6, 7, 22].

EFSO (Efficient solution) [12]: A schedule α^* is known as efficient solution or Pareto optimal (non-dominated). If another schedule α satisfying $h_j(\alpha) \leq h_j(\alpha^*), j = 1,2,..,n$ cannot be found, considering that at least one of the aforementioned is a strict disparity. Another way is that α^* is dominated by α [20].

F_{CET} : OF of $(S_{CE}M_T)$ -problem, and F_{SP} is objective function of (SP)-problem.

Feasible schedule: Any schedule $\beta \in S$ (S is the collection of all schedules) can be considered feasible if it meets the problem's constraints.

MST: Jobs are arranged according to their slack time $s_j = d_j - p_j$ in a non-decreasing order (where $s_1 \leq s_2 \leq \cdots \leq s_n$). For minimizing E_{max} with the use of this rule [8].

MOF: Multi-objective function;

N: Number

MCF: Multi- criteria function.

NEFS: Number of efficient solutions.

n_i: N. 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 optimal in the case when there isn't other schedule σ that satisfies $f_j(\sigma) \leq f_j(\sigma^*)$, where j from 1 to k(k: N. 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^*.

Ver: $0 < \text{Veritable} < 1$.

SPT: The jobs are being processed in a non-descending order p_j (i.e. $p_1 \leq p_2 \leq \cdots \leq p_n$), it is commonly known that this rule minimizes $\sum C_j$ for the $1//\sum C_j$ problem.

The mathematical model for the $1//\left(\sum C_j, \sum E_j, T_{max}\right)$ problem

The problem aims to discover an efficient solution that yields the minimal value of the tri-criteria. Total completion time $\sum C_j$, total earliness time $\sum E_j$, and the maximum tardiness T_{max}; this problem is denoted by:

$$\left.\begin{aligned} & F_{CET} = Min\left(\sum C_j, \sum E_j, T_{max}\right) \\ & \quad \text{subject to} \\ & \left.\begin{aligned} & C_j \geq p_j(\sigma), \\ & C_j = \sum_{k=1}^{j-1} p_k(\sigma) + p_j(\sigma), \\ & T_j \geq C_j - d_j(\sigma), \\ & E_j \geq d_j(\sigma) - C_j, \\ & E_j \geq 0, \text{and } T_j \geq 0, \end{aligned}\right\} j \text{ from 1 to } n \end{aligned}\right\} \tag{1}$$

This problem is referred to as the $(S_{CE}M_T)$ -problem.

For $(S_{CE}M_T)$ -problem, sub-problem can be concluded that $1//\left(\sum C_j + \sum E_j + T_{max}\right)$ problem that referred to the (SP)-problem, and it can be defined as follows:

$$\left.\begin{aligned} & F_{SP} = Min\left(\sum C_j + \sum E_j + T_{max}\right) \\ & \quad \text{subject to} \\ & \left.\begin{aligned} & C_1 = p_1(\sigma), \\ & C_j = \sum_{k=1}^{j-1} p_k(\sigma) + p_j(\sigma), \\ & T_j \geq C_j - d_j(\sigma), \\ & E_j \geq d_j(\sigma) - C_j, \\ & E_j \geq 0, \text{and } T_j \geq 0, \end{aligned}\right\} j \text{ from 1 to } n \end{aligned}\right\} \tag{2}$$

3. METHODOLOGY

In this section, two exact method (BAB and CEM) and two HMs were introduced for solving the $(S_{CE}M_T)$-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 HMs were proposed and were adopted to find efficient solutions to this problem in a reasonable time.

3.1 Exact method

We have presented two exact methods in this subsection (CEM and BAB). The CEM was used as a simple approach that generates all of the feasible tables for choosing the optimal solution. While, 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 by methodically reviewing subsets of potential solutions. A search tree with nodes corresponding to these subsets has been utilized for describing BAB.

3.1.1 BAB method to solve the $(S_{CE}M_T)$-problem

In this subsection, two BAB techniques will be used to solve this problem.

First technique is BAB without DRs (BAB(WODRs)). This method can be summarized as follows: the 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 steps for BAB(WODRs) can be seen below:

Algorithm 1: BAB(WODRs) Algorithm

Step 1: Enter: n, p_j & d_j, where j from 1 to n.
Step 2: Let $\mathcal{S} = \varphi$, for any α define $F_{CET}(\alpha) = \left(\sum C_j(\alpha), \sum E_j(\alpha), T_{max}(\alpha)\right)$.
Step 3: Calculate an upper bound (UB) of $(S_{CE}M_T)$-problem, according to the order of jobs in $\alpha =$ MST. Let $\text{UB}_{CET} = F_{CET}(\alpha) = \left(\sum C_j(\alpha), \sum E_j(\alpha), T_{max}(\alpha)\right)$ at the search tree's parent node, where $j = 1, 2, \ldots, n$.
Step 4: For every node in the BAB approach's search tree and each partial sequence σ of jobs, compute LB(σ)= The objective function's cost of sequencing jobs in σ + the cost of un-sequenced jobs arranged according to the SPT rule (where $\sigma =$ SPT).
Step 5: A branch of each node with LB does not dominate the UB.
Step 6: Obtaining a set of solutions at the final level of the search tree, if $F(\sigma)$ the result is indicated, σ is added to the set $\mathcal{S}$ unless they are dominated by efficient solutions that have been obtained previously in $\mathcal{S}$, this process is called filtering $\mathcal{S}$.
Step 7: End.

Second technique is BAB with DRs (BAB(WDRs)). This method could be summarized as follows: The UB and LB of each node for the un-sequenced portion will be based on the SPT rule. To decrease the number of open nodes, which saves time and increases the number of solved problems, this BAB depends on DR, since the size of search tree (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 goal of dominance rules is for reducing the available research on scheduling problems. Consequently, as a process for reducing search area and shorten search period. Several Dominance Rules can be used to reduce the current sequence. DRs typically indicate some (all) sections of the path in order to acquire a good value for the objective function, and they can be valuable in determining if a node in the BAB method can be discarded before its lower bound (LB) is determined. DRs are clearly useful when a node can be ignored despite having a less-than-optimal LB. The DRs are also useful in the BAB approach for eliminating nodes that are dominated by others. These enhancements result in a significant reduction in the number of nodes required to achieve the efficient (optimal) solution. By applying the following theorem:

Theorem (1) [23]: 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 is processed before the job k.

Algorithm 2: BAB(WDRs) Algorithm

Step 1: Enter: n, p_j & d_j, where j from 1 to n. Find adjacency matrix A.
Step 2: Let $\mathcal{S} = \varphi$, for any α define $F_{CET}(\alpha) = \left(\sum C_j(\alpha), \sum E_j(\alpha), T_{max}(\alpha)\right)$.
Step 3: Calculate an upper bound (UB) of $(S_{CE}M_T)$ problem, by arranging jobs in $\alpha =$ SPT. Calculate the $\text{UB}_{CET} = F_{CET}(\alpha) = \left(\sum C_j(\alpha), \sum E_j(\alpha), T_{max}(\alpha)\right)$ at the search tree's parent node, where $j = 1, 2, \ldots, n$.
Step 4: For every node in the BAB approach's search tree and each partial sequence σ of jobs, compute LB(σ)= The objective function's cost of sequencing jobs in σ + the cost of un-sequenced jobs arranged according to the $\sigma =$ SPT rule.
Step 5: Branch from every node within LB $\leq$ UB and check $i \to j$.
Step 6: Obtaining a set of solutions at the final level of the search tree; 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, the same BAB that used for the $(S_{CE}M_T)$-problem with some modifications indicated by BAB. First, calculate UB for (SP)-problem s.t. UB($\alpha =$ SPT) $= F_{SP}(\alpha) = \sum C_j(\alpha) + \sum E_j(\alpha) + T_{max}(\alpha)$, then compute the LB of any node comprising of sequence and un-sequence components (by SPT rule). s.t. LB($\sigma =$ SPT) $= F_{SP}(\sigma) = \sum C_j(\sigma) + \sum E_j(\sigma) + T_{max}(\sigma)$, where σ is the rule for un-sequenced jobs. Repeat these steps until an optimal solution is obtained from the root.

3.2 HMs for $(S_{CE}M_T)$-problem and (SP)-problem

Heuristic methods speed up the process of reaching a satisfactory solution. Many researchers have used heuristic algorithms to solve NP-hard problems [24]. In this subsection, two heuristic algorithms were proposed (SM-$(S_{CE}M_T)$ and DR-$(S_{CE}M_T)$ for solving the $(S_{CE}M_T)$-problem and the (SP)-problem:

3.2.1 SM-$(S_{CE}M_T)$ method

SM-$(S_{CE}M_T)$ method is proposed for solving $(S_{CE}M_T)$-

problem and (SP)-Problem in this subsection [25]. Firstly, the objective function using the SPT rule is calculated. Next, arrange the third job in the second position, with the other jobs arranged in accordance with the SPT rule and compute OF, etc., up to n sequences are obtained, then repeat the same procedures when using the MST rule, as described below:

Algorithm 3: SM-$(\boldsymbol{S_{CE}M_T})$ Heuristic Algorithm

Step 1: Enter: n, p_j & d_j, where j from 1 to $n, \mathcal{S} = \varphi$.
Step 2: Place the jobs in SPT rule order (β_1) and compute $F_{11}(\beta_1) = \left(\sum C_j(\beta_1), \sum E_j(\beta_1), T_{max}(\beta_1)\right)$; $\mathcal{S} = \mathcal{S} \cup \{F_{11}(\beta_1)\}$.
Step 3: Place job i at the first position of β_{i-1} from $i = 2$ to n to get β_i and compute $F_{1i}(\beta_i) = \left(\sum C_j(\beta_i), \sum E_j(\beta_i), T_{max}(\beta_i)\right)$; $\alpha = \alpha \cup \{F_{1i}(\beta_i)\}$.
End;
Step 4: Place the jobs in MST rule order (σ_1) and compute calculate $F_{21}(\sigma_1) = \left(\sum C_j(\sigma_1), \sum E_j(\sigma_1), T_{max}(\sigma_1)\right)$; $\mathcal{S} = \mathcal{S} \cup \{F_{21}(\sigma_1)\}$.
Step 5: Place job i at the first position of (σ_{i-1}) from $i = 2$ to n to get σ_i and compute $F_{2i}(\sigma_i) = \left(\sum C_j(\sigma_i), \sum E_j(\sigma_i), T_{max}(\sigma_i)\right)$; $\mathcal{S} = \mathcal{S} \cup \{F_{2i}(\sigma_i)\}$.
End;
Step 6: To find a set of efficient solutions for $(S_{CE}M_T)$ problem, filter set $\mathcal{S}$.
Step 7: Output: $\mathcal{S}$ represents a set of efficient solutions.
Step 8: End.

3.2.2 DR- $(S_{CE}M_T)$ heuristic method

DR- $(S_{CE}M_T)$ depends on DRs is proposed for solving $(S_{CE}M_T) -$problem and (SP)-problem. To summarize DR-$(S_{CE}M_T)$ method, find a sequence sort with a minimum of p_j and dj, corresponding to the DRs, and compute the objective function. DR- $(S_{CE}M_T)$ algorithm is summarized in the following steps:

Algorithm 4: DR-$(\boldsymbol{S_{CE}M_T})$ Heuristic Algorithm

Step 1: Enter: n, p_j & d_j, where j from 1 to n.
Step 2: Employ 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: Discover the sequence α_1 with a non-increasing order of p_j that does not conflict together with matrix A (DR); if $p_j = p_i$,where $j, i \in N$ then order α_1 by d_j, then $\mathcal{S} = \mathcal{S} \cup \{\alpha_1\}$.
Step 4: Discover the sequence α_2 with a non-increasing order of d_j that does not conflict together with matrix A (DR); if $d_j = d_i$,where $j, i \in N$ then order α_2 by d_j, then $\mathcal{S} = \mathcal{S} \cup \{\alpha_2\}$.
Step 5: Determine the set of the dominant sequence $\mathcal{S}'$ from $\mathcal{S}$.
Step 6: Compute $F_{CET}(\mathcal{S}')$.
Step 7: Output: $\mathcal{S}'$ (the set of effective solution)
Step 8: End.

4. RESULTS AND DISCUSSION

This section, considered compaction results of Exact and HMs to the $(S_{CE}M_T)$-problem and (SP)-problem. Because we deal with the MSP, the p_j and d_j values are randomly generated for five examples s.t., $p_j \in [1,10]$ and: $d_j \in \begin{cases} [1,30] & 1 \le n \le 29 \\ [1,40] & 30 \le n \le 99 \\ [1,50] & 100 \le n \le 999' \\ [1,70] & \text{otherwise} \end{cases}$
subject to condition $d_j \ge p_j$, for $j = 1, 2, \dots, n$.

4.1 Results and discussion of the $(\boldsymbol{S_{CE}M_T})$-problem

In this subsection, the results of applying the exact methods will be compared with the heuristic methods for the problem $(S_{CE}M_T)$. All results from using all presented methods are averages of five examples for each n. Table 1 illustrated the comparison results of BAB(WODR), BAB(WDR), and CEM for the problem $(S_{CE}M_T)$ with $n = 4, 5, \dots, 11$. In Table 2, the results of BAB without and with DR for problem $(S_{CE}M_T)$, where $n = 12: 19,20,30,40,50$ were presented. Also, Table 3 showed the comparison results between the proposed heuristics methods (SM-$(S_{CE}M_T)$ and DR-$(S_{CE}M_T)$), with CEM for the problem $(S_{CE}M_T)$ with $n = 4$ to 11. In addition, the results of SM-$(S_{CE}M_T)$ and DR-$(S_{CE}M_T)$ that were compared to the BAB(WODR), and BAB(WDR) for the problem $(S_{CE}M_T)$ have been listed in Table 4, for different values of n. Table 5 presented the results of SM-$(S_{CE}M_T)$ and DR-$(S_{CE}M_T)$ for problem $(S_{CE}M_T)$ for different n.

The simulation result for exact and heuristic methods for solving the $1//\left(\sum C_j, \sum E_j, T_{max}\right)$ problem can be analyzed in terms accuracy and CPU-Time. From Table 1 illustrated that CEM gave minimum values for the $1//\left(\sum C_j, \sum E_j, T_{max}\right)$ problem compared to the results of the BAB up to $n \le 11$. Also, CEM was taken a long time (CPU-Time) compared to BAB. In additionally, the BAB(WODR) starts to give the minimum values for the $(S_{CE}M_T)$ problem compared to the results for BAB(WDR) for $n \le 7$, while BAB(WDR) starts to give the minimum values for the $(S_{CE}M_T)$ problem compared to the results for BAB(WODR) for $n > 7$.

Moreover, from Table 2, BAB(WDR) gave minimum values in terms accuracy and CPU-Time compared with results of BAB(WODR), for $n = 12$ to 50. Therefore, BAB(WDR) performs better than BAB(WODR), and BAB without DR solved the problem in all cases from $n = 12$ to 19, but failed to solve the problem when $n \ge 19$, whereas BAB with DR (BAB(WDR)) solved the problem in all cases from $n = 12$ to 55, but failed to solve the problem when $n > 50$. Moreover, from Tables 1 and 2, the results of the BAB with dominance rules confirmed the number of efficient solutions to the problems less than the number of efficient solutions for BAB without dominance rules and CEM. In addition, from Table 3, the CEM gave best results compared to the heuristic methods (SM-$(S_{CE}M_T)$ and DR-$(S_{CE}M_T)$. Furthermore, CEM takes a long time in the CPU-Time, whereas the HM SM-$(S_{CE}M_T)$ gives better results than DR-$(S_{CE}M_T)$ for $n \le 11$. In Table 4, BAB(WDR) gave best showed results when compared with BAB (WODR), SM-$(S_{CE}M_T)$, and DR-$(S_{CE}M_T)$ which also showed that BAB without DR solved all cases of the problem from $n = 4$ to $n = 19$, and BAB with DR failed to solve all problems for $n > 50$. In general, the results of BAB method are better when compared to Heuristic Methods up to $n = 50$. Also, heuristic method SM-$(S_{CE}M_T)$ gives better results than $\mathrm{DR} - (S_{CE}M_T)$ for $n \le 50$. While $\mathrm{DR} - (S_{CE}M_T)$ gives better results than SM-$(S_{CE}M_T)$ for $50 < n \le 5000$, for problem $(S_{CE}M_T)$. From Table 5, the heuristic method (SM-($S_{CE}M_T$)) gave better result than DR-($S_{CE}M_T$) from $n = 4$ to 60 while heuristics method DR-($S_{CE}M_T$) gave better results than SM-($S_{CE}M_T$) for problem ($S_{CE}M_T$) up to $60 < n \leq 5000$. Furthermore, DR-($S_{CE}M_T$) was successful in resolving all issues for $n \leq 4000$ and unable to resolve every problem for $n > 4000$, whereas SM-($S_{CE}M_T$) was successful in resolving all issues for $n \leq 5000$ and didn't manage to resolve every problem for $n > 5000$.

Table 1. Comparison between BAB(WODR) and BAB(WDR) with CEM for problem ($S_{CE}M_T$), for $n = 4, 5, \ldots, 11$

EX	CEM			BAB(WODR)LB=SPT, UB=MST			BAB(WDR)LB=SPT, UB=MST		
	MCF	TIME	NES	MCF	TIME	NES	MCF	TIME	NES
n_5	AV(F_{CET})	ACT_S	AN_{EFS}	AV(F_{CET})	ACT_S	AN_{EFS}	AV(F_{CET})	ACT_S	AN_{EFS}
4	(60.8,24.2,2.2)	Ver	8.2	(60.7,24.4,2.2)	Ver	8.0	(59.0,27.3,1.8)	Ver	4.6
5	(90.1,23.3,6.5)	Ver	10.2	(90.1,23.3,6.5)	Ver	10.2	(100.6,20.3,10.1)	Ver	4.8
6	(112.1,24.6,9.8)	Ver	14.6	(112.1,24.8,9.8)	Ver	20.6	(120.5,23.6,11.2)	Ver	12.2
7	(125.5,23.1,11.8)	2.09	36	(124.3,23.8,12.1)	Ver	28.6	(126.3,22.4,12.6)	Ver	22.6
8	(153.3,26.1,16.1)	45.5	56.2	(152.3,26.4,16.9)	Ver	39.2	(149.8,29.5,16.2)	Ver	31.2
9	(215.1,19.0,24.7)	985.4	35.4	(216.6,19.1,25.5)	Ver	30.4	(182.5,20.8,17.9)	Ver	24.8
10	(205.0,18.3,12.1)	87.2	118.6	(209.8,35.0,21.8)	Ver	92.4	(193.0,38.1,18.1)	Ver	71.8
11	(301.0,35.5,8.3)	1800	72.4	(294.2,22.8,37.8)	Ver	34.4	(286.6,23.9,35.3)	Ver	29.8

Table 2. Comparison between the BAB without and with DR for problem ($S_{CE}M_T$)

EX	BAB(WODR)LB=SPT, UB=MST			BAB(WDR)LB=SPT, UB=MST		
	MCF	TIME	NES	MCF	TIME	NES
n_5	AV(F_{CET})	ACT_S	AN_{EFS}	AV(F_{CET})	ACT_S	AN_{EFS}
12	(345.7,28.2,41.9)	Ver	50.6	(323.7,28.2,37.7)	Ver	39.2
13	(357.9,21.6,43.6)	1.2	58.8	(352.2,23.5,43.9)	Ver	48.4
14	(470.8,11.9,59.9)	Ver	29.4	(458.4,14.8,58.2)	Ver	20.4
15	(549.8,23.1,65.6)	3.6	56.6	(556.8,25.4,66.8)	Ver	48.6
16	(529.9,28.2,60.6)	3.6	61.6	(514.4,31.1,60.9)	Ver	47.0
17	(633.3,20.7,72.7)	39.9	38.8	(631.9,21.4,72.7)	Ver	33.2
18	(708.6,37.0,75.2)	57.9	83.6	(702.0,38.0,74.0)	Ver	79.6
19	(738.2,28.7,77.5)	93.5	54.2	(697.1,33.4,75.7)	Ver	48.2
20	-	-	-	(827.9,35.5,81.7)	Ver	55.4
30	-	-	-	(1983.5,26.0,148.5)	Ver	32.2
40	-	-	-	(3351.8,36.4,204.8)	3.0	49.2
50	-	-	-	(4950.1,36.8,246.8)	31.8	56.8

Table 3. Comparison of SM-($S_{CE}M_T$), DR-($S_{CE}M_T$) and CEM of problem ($S_{CE}M_T$), $n = 4, 5, \ldots, 11$

EX	CEM			SM-($S_{CE}M_T$)			DR-($S_{CE}M_T$)		
	MCF	TIME	NES	MCF	TIME	NES	MCF	TIME	NES
n_5	AV(F_{CET})	ACT_S	AN_{EFS}	AV(F_{CET})	ACT_S	AN_{EFS}	AV(F_{CET})	ACT_S	AN_{EFS}
4	(60.8,24.2,2.2)	Ver	8.2	(60.2,25.7,2.9)	Ver	5.6	(61.7,25.3,4.0)	Ver	4.8
5	(90.1,23.3,6.5)	Ver	10.2	(91.9,24.4,7.3)	Ver	5.8	(92.9,24.9,9.3)	Ver	5.0
6	(112.1,24.6,9.8)	Ver	21	(113.7,28.9,11.3)	Ver	6.8	(116.6,26.9,12.9)	Ver	4.8
7	(125.5,23.1,11.8)	2.09	32.6	(131.4,24.1,14.4)	Ver	7.8	(133.7,24.1,15.6)	Ver	5.6
8	(153.3,26.1,16.1)	45.5	56.2	(155.6,32.4,18.0)	Ver	8.0	(157.0,31.8,18.4)	Ver	5.6
9	(215.1,19.0,24.7)	985.4	35.4	(225.3,21.3,27.7)	Ver	6.8	(232.7,23.0,29.2)	Ver	4.6
10	(205.0,18.3,12.1)	87.2	40	(224.5,36.5,23.0)	Ver	10.4	(227.5,35.0,23.7)	Ver	6.2
11	(301.0, 35.5, 8.3)	1800	26.8	(317.7,23.9,37.9)	Ver	8.8	(326.7,22.8,38.9)	Ver	6.2

Table 4. Comparison results SM-($S_{CE}M_T$)and DR-($S_{CE}M_T$)with BAB (WODR), and BAB (WDR) for problem ($S_{CE}M_T$)

EX	BAB(WODR), LB=SPT, UB=MST		BAB(WDR), LB=SPT, UB=MST		SM-($S_{CE}M_T$)		DR-($S_{CE}M_T$)	
	MCF	TIME	MCF	TIME	MCF	TIME	MCF	TIME
n_5	AV(F_{CET})	ACT_S	AV(F_{CET})	ACT_S	AV(F_{CET})	ACT_S	AV(F_{CET})	ACT_S
12	(345.7,28.2,41.9)	Ver	(323.7,28.2,37.7)	Ver	(378.9,31.8,42.9)	Ver	(396.8,27.8,46.8)	Ver
13	(357.9,21.6,43.6)	1.2	(352.2,23.5,43.9)	Ver	(415.5,31.8,44.2)	Ver	(429.3,30.3,47.2)	Ver
14	(470.8,11.9,59.9)	Ver	(458.4,14.8,58.2)	Ver	(507.7,15.1,58.4)	Ver	(515.8,14.3,58.4)	Ver
15	(549.8,23.1,65.6)	3.6	(556.8,25.4,66.8)	Ver	(624.7,26.4,64.8)	Ver	(641.9,22.5,65.7)	Ver
16	(529.9,28.2,60.6)	3.6	(514.4,31.1,60.9)	Ver	(601.3,32.4,60.4)	Ver	(628.9,30.8,62.4)	Ver
17	(633.3,20.7,72.7)	39.9	(631.9,21.4,72.7)	Ver	(711.5,24.3,72.9)	Ver	(739.1,21.8,72.3)	Ver
18	(708.6,37.0,75.2)	57.9	(702.0,38.0,74.0)	Ver	(832.8,40.6,75.2)	Ver	(859.7,35.1,76.8)	Ver

EX	BAB(WODR), LB=SPT, UB=MST		BAB(WDR), LB=SPT, UB=MST		SM-($S_{CE}M_T$)		DR-($S_{CE}M_T$)	
	MCF	TIME	MCF	TIME	MCF	TIME	MCF	TIME
n_5	$AV(F_{CET})$	ACT_S	$AV(F_{CET})$	ACT_S	$AV(F_{CET})$	ACT_S	$AV(F_{CET})$	ACT_S
19	(738.2,28.7,77.5)	93.5	(697.1,33.4,75.7)	Ver	(880.3,30.2,79.3)	Ver	(917.8,24.7,79.9)	Ver
20	-	-	(827.9,35.5,81.7)	Ver	(1039.4,31.3,89.1)	Ver	(1090.8,20.6,92.9)	Ver
30	-	-	(1983.5,26.0,148.5)	Ver	(1513.0,45.3,106.9)	Ver	(2262.2,23.8,143.8)	Ver
40	-	-	(3351.8,36.4,204.8)	3.0	(4022.5,32.7,201.1)	Ver	(4036.1,31.6,203)	Ver
50	-	-	(4950.1,36.8,246.8)	31.8	(6045.2,37.5,246.0)	Ver	(6116.3,30.4,245.3)	Ver
100	-	-	-	-	(25917.9,38.9,544.9)	Ver	(24137.7,24.5,547.2)	Ver
1000	-	-	-	-	(2637870,32,5487)	5.7	(1935453.8,0,5484.6)	9.9
2000	-	-	-	-	(10582517.5,35.6,10983.5)	26.8	(7713263,0,10980.6)	72.3
3000	-	-	-	-	(23701295.9,30.0,16449.5)	80.4	(17314163.2,0.0,16447)	264.9
4000	-	-	-	-	(42269531.3,36.5,22023.2)	183.6	(30915011.2,0.0,22020)	647.8
5000	-	-	-	-	(65964191.8,29.1,27429.7)	341.4	(48048599.0,0.0,27427)	1263.1

Table 5. Comparison results SM-($S_{CE}M_T$)and DR-($S_{CE}M_T$)for problem ($S_{CE}M_T$)

EX	SM-($S_{CE}M_T$)			DR-($S_{CE}M_T$)		
	MCF	TIME	NES	MCF	TIME	NES
n_5	AV(F_{CET})	ACT_S	AN_{EFS}	AV(F_{CET})	ACT_S	AN_{EFS}
40	(4022.5,32.7,201.1)	Ver	16.4	(4036.1,31.6,203.0)	Ver	10.6
60	(8577.7,32.9,298.0)	Ver	18.2	(8605.1,24.5,299.6)	Ver	11.8
80	(16483.9,37.1,424.9)	Ver	20.8	(15740.9,31.6,426.9)	Ver	13.8
100	(25917.9,38.9,544.9)	Ver	21.0	(24137.7,24.5,547.2)	Ver	12.4
400	(415738.1,34.3,2179.2)	1.2	34.2	(330938.5,1.8,2176.6)	1.4	5.0
600	(940295.2,35.0,3258.7)	2.3	38.2	(714625.0,0.1,3257.1)	3.1	1.4
800	(1688582.7,29.4,4394.1)	3.8	39.0	(1263913. 4,0.1,4390.8)	5.9	1.2
1000	(2637870.0,32.0,5487.0)	5.8	39.2	(1935453.8,0.0,5484.6)	10.1	1.0
2000	(10582517.5,35.6,10983.5)	27.3	40.2	(7713263.0,0.0,10980.6)	74.0	1.0
3000	(23701295.9,30.0,16449.5)	80.4	40.4	(17314163.2,0.0,16447.8)	264.9	1.0
4000	(42269531.3,36.5,22023.2)	183.6	40.4	(30915011.2,0.0,22020.2)	647.8	1.0
5000	(65964191.8,29.1,27429.7)	341.4	39.6	(48048599.0,0.0,27427.0)	1263.1	1.0

4.2 Results and discussion of the (SP)-problem

In this subsection, the simulation results of the exact methods will be compared with the two suggested heuristic methods with regard to the subproblem (SP), the outcomes of the comparison were presented in Tables 6-8.

In Table 6, the results of BAB(WODR), and BAB(WDR) methods were compared with the CEM for the problem (SP) in different values of n ($n = 4$ to 17). This table illustrated that CEM gave same results of BAB(WODR) and better than BAB(WDR), but CEM taken more time, and BAB(WODR) takes a long time in CUP-Time compared to BAB(WDR). Moreover, CEM solved the problem when $n \leq 11$, BAB(WODR) solved the problem when $4 \leq n \leq 15$, BAB(WDR) solved the problem when $4 \leq n \leq 17$.

In Table 7, the results of SM-(SP) and DR-(SP) were compared with CEM for the problem (SP) are presented for $n = 4$ to 11. Furthermore, Table 7 shows that result of CEM best from the SM-(SP) and DR-(SP). However, CEM takes a long time compared to SM-(SP)and DR-(SP) for the problem (SP). For better illustration for Table 7, all the comparative results between SM-(SP) and DR-(SP), and CEM for problem (SP) on different values of n ($n = 4, 5, \ldots, 11$) are depicted in Figure 1.

Table 6. Comparison results of BAB(WODR) and BAB(WDR) with CEM for subproblem (SP)

EX	CEM		BAB(WODR), LB=UB=SPT		BAB(WDR), UB=LB=SPT	
	MOF	TIME	MOF	TIME	MOF	TIME
n_5	AV(F_{SP})	ACT_S	AV(F_{SP})	ACT_S	AV(F_{SP})	ACT_S
4	84.2	Ver	84.2	Ver	84.2	Ver
5	116.6	Ver	116.6	Ver	116.6	Ver
6	141	Ver	141.0	Ver	141.0	Ver
7	154.2	Ver	154.2	Ver	154.2	Ver
8	189.6	Ver	189.6	Ver	189.6	Ver
9	253.8	6.7	253.8	Ver	253.8	Ver
10	257.6	71.4	257.6	351.4	257.6	Ver
11	348	833.3	348.0	145.8	348.0	Ver
12	-	-	409.8	628.2	409.8	Ver
13	-	-	418.2	1800	418.2	Ver
14	-	-	537.2	254.3	537.2	Ver
15	-	-	629	1800	629.6	86.3
16	-	-	-	-	607.4	16.7
17	-	-	-	-	718.2	0.9

Table 7. Comparison between SM-(SP) and DR-(SP) with CEM for problem (SP)

EX	CEM		SM-(SP)		DR-(SP)	
	MOF	TIME	MOF	TIME	MOF	TIME
n_5	AV(F_{SP})	ACT_S	AV(F_{SP})	ACT_S	AV(F_{SP})	ACT_S
4	84.2	Ver	84.2	Ver	85.6	Ver
5	116.6	Ver	118.2	Ver	117.2	Ver
6	141	Ver	147.2	Ver	144.0	Ver
7	154.2	Ver	162.8	Ver	163.8	Ver
8	189.6	Ver	194.0	Ver	196.6	Ver
9	253.8	6.7	267.0	Ver	267.6	Ver
10	257.6	71.4	272.4	Ver	270.4	Ver
11	348	833.3	358.4	Ver	361.4	Ver

Table 8. Comparison results between the BAB without DR, BAB with DR, SM-(SP), and DR-(SP) for the problem (SP)

EX	BAB(WODR), LB=UB=SPT		BAB(WDR), UB=LB=SPT		SM-(SP)		DR-(SP)	
	MOF	TIME	MOF	TIME	MOF	TIME	MOF	TIME
n_5	AV(F_{SP})	ACT_S	AV(F_{SP})	ACT_S	AV(F_{SP})	ACT_S	AV(F_{SP})	ACT_S
4	84.2	Ver	84.2	Ver	84.2	Ver	85.6	Ver
5	116.6	Ver	116.6	Ver	118.2	Ver	117.2	Ver
6	141.0	Ver	141.0	Ver	147.2	Ver	144.0	Ver
7	154.2	Ver	154.2	Ver	162.8	Ver	163.8	Ver
8	189.6	Ver	189.6	Ver	194.0	Ver	196.6	Ver
9	253.8	Ver	253.8	Ver	267.0	Ver	267.6	Ver
10	257.6	351.4	257.6	Ver	272.4	Ver	270.4	Ver
11	348.0	145.8	348.0	Ver	358.4	Ver	361.4	Ver
12	409.8	628.2	409.8	Ver	423.4	Ver	426.2	Ver
13	418.2	1800	418.2	Ver	450.4	Ver	451.2	Ver
14	537.2	254.3	537.2	Ver	548.0	Ver	545.6	Ver
15	629	1800	629.6	86.3	648.2	Ver	648.4	Ver
16	-	-	607.4	16.7	625.2	Ver	623.8	Ver
17	-	-	718.2	Ver	738.6	Ver	734.8	Ver
18	-	-	-	-	828.6	Ver	828.2	Ver
19	-	-	-	-	854.8	Ver	851.0	Ver
20	-	-	-	-	1031.0	Ver	1030.0	Ver
40	-	-	-	-	3583.2	Ver	3580.4	Ver
60	-	-	-	-	7359.0	Ver	7346.8	Ver
80	-	-	-	-	13638.8	Ver	13616.0	Ver
100	-	-	-	-	21258.8	Ver	21236.8	Ver
400	-	-	-	-	312938.6	Ver	312795.4	Ver
600	-	-	-	-	697160.0	1.1	697006.6	1.3
800	-	-	-	-	1245389.8	1.8	1245259.0	2.5
1000	-	-	-	-	1941079.4	2.7	1940938.4	4.1
2000	-	-	-	-	7724404.0	13.2	7724243.6	27.8
3000	-	-	-	-	17330742.8	39.2	17330611.0	104.8
4000	-	-	-	-	30937191.0	86.2	30937031.4	248.7

Figure 1. The comparison results between SM-(SP) and DR-(SP) with CEM, $n = 4, 5, \dots, 11$

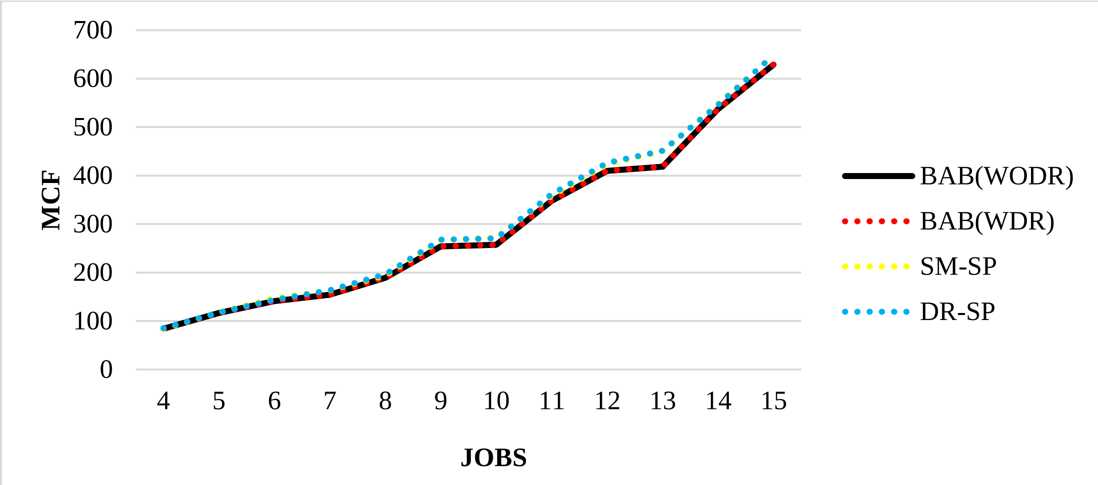

Figure 2. Results of the comparison between the BAB with and without DR, SM-(SP), and DR-(SP) for problem (SP)

Table 8 presents the results of heuristic methods (SM-(SP) and DR-(SP), and Exact methods (BAB without and with DR) for problem (SP) for different value of n ($n = 4$ to 20,40,60,80,100,400,600,800,1000,2000,3000, and 4000). In addition, by using Table 8, Figure 2 shows the comparison result between BAB and HM for the problem (SP). Moreover, in Table 8, BAB(WODR) gave best s results when compared with BAB (WDR), SM-(SP) and DR-(SP), which also showed that BAB with DR solved all problem situations from $n = 4$ to 17. In addition, all instances of the problem were solved by BAB without DR, from $n = 4$ to 15, and when $n >$ 15, was unable to solve the problems. However, SM-(SP) and DR-(SP) solved all the problems from $n = 4$ to $n = 4000$, but the BAB (WODR) and BAB (WDR) method gave better results.

5. CONCLUSIONS

In this paper, two new techniques two new Heuristic methods SM-$(S_{CE}M_T)$ and DR-$(S_{CE}M_T)$ were proposed to solve the tri-criteria problem (1//$\left(\sum C_j, \sum E_j, T_{max}\right)$), three multi objective (1// $\left(\sum C_j + \sum E_j + T_{max}\right)$) machine scheduling problems. In addition, BAB with DRs, BAB without DRs, and CEM as an Exact method were used to compare results n terms of accuracy and computational time. The result showed that, SM-$(S_{CE}M_T)$ performs better than DR-$(S_{CE}M_T)$ were for all $n \leq 400$, while, when $n \geq 400$, DR-$(S_{CE}M_T)$ gave better results than SM-$(S_{CE}M_T)$. Furthermore, for two problems, the result of the BAB with dominance rules for two problems showed a lower number of efficient solutions for all n. than the number of efficient solutions of d BAB without dominance rules and CEM. For future work, a new UB and LB can be used for the BAB algorithm to prove its effectiveness in determining the best solution for the MOF. Different machine environments can be used to study more complex problems and/or our proposed problems can be completed with constraints, such as the release date (r_j), setup time (S_f), and pre-stopping.

ACKNOWLEDGMENT

The authors would like to thank Baghdad University at (www.uobaghdad.edu.iq) Baghdad – Iraq for its support in the present work.

REFERENCES

[1] Ali, F.H., Jawad, A.A. (2020). Minimizing the total completion time and total earliness time functions for a machine scheduling problem using local search methods. Iraqi Journal of Science, 126-133. https://doi.org/10.24996/ijs.2020.SI.1.17

[2] Akande, S., Ajisegiri, G.O., Adegoke, A.A., Ikumapayi, O.M., Akinlabi, E.T. (2022). Dispatching rules for minimizing deviation from JIT schedule using the earliness-tardiness scheduling problem with due windows approach. Journal Européen des Systèmes Automatisés, 55(3): 377-385. https://doi.org/10.18280/jesa.550310

[3] Channappa, A.K., Ramaiah, N., Rajanna, S.B. (2022). Multi-objective optimization method for task scheduling and resource allocation in cloud environment. Revue d'Intelligence Artificielle, 36(2): 223-232. https://doi.org/10.18280/ria.360206

[4] Smith, W.E. (1956). Various optimizers for single-stage production. Naval Research Logistics Quarterly, 3(1-2): 59-66. https://doi.org/10.1002/nav.3800030106

[5] Hoogeveen, J.A. (1996). Single-machine scheduling to minimize a function of two or three maximum cost criteria. Journal of Algorithms, 21(2): 415-433. https://doi.org/10.1006/jagm.1996.0051

[6] Hall, L.A., Shmoys, D.B. (1992). Jackson's rule for single-machine scheduling: Making a good heuristic better. Mathematics of Operations Research, 17(1): 22-35. http://www.jstor.org/stable/3689890.

[7] Lageweg, B.J., Lenstra, J.K., Kan, A.R. (1976). Minimizing maximum lateness on one machine: Computational experience and some applications. Statistica Neerlandica, 30(1): 25-41. https://doi.org/10.1111/j.1467-9574.1976.tb00264.x

[8] Hoogeveen, J.A. (1996). Minimizing maximum promptness and maximum lateness on a single machine. Mathematics of Operations Research, 21(1): 100-114. https://doi.org/10.1287/moor.21.1.100

[9] Abdul-Razaq, T.S., Motair, H.M. (2017). Solving composite multi objective single machine scheduling

problem using branch and bound and local search algorithms. Al-Mustansiriyah Journal of Science, 28(3): 200-208. http://doi.org/10.23851/mjs.v28i3.122

[10] Atiya, B., Bakheet, A.J.K., Abbas, I.T., Bakar, M., Abu, R., Soon, L.L., Monsi, M.B. (2016). Application of simulated annealing to solve multi-objectives for aggregate production planning. In AIP Conference Proceedings, 1739(1): 4952566. https://doi.org/10.1063/1.4952566

[11] Abdul-Razaq, T.S., Ali, Z.M. (2017). Minimizing the total completion times, the total tardiness and the maximum tardiness. Ibn AL-Haitham Journal for Pure and Applied Science, 28(2): 155-170. https://doi.org/10.30526/36.1.2994

[12] Jawad, A.A., Ali, F.H., Hasanain, W.S. (2020). Using heuristic and branch and bound methods to solve a multi-criteria machine scheduling problem. Iraqi Journal of Science, 2055-2069. https://doi.org/10.24996/ijs.2020.61.8.21

[13] Ahmed, M.G., Ali, F.H. (2022). Exact method with dominance rules for solving scheduling on a single machine problem with multiobjective function. Al-Mustansiriyah Journal of Science, 33(2): 56-63. https://doi.org/10.23851/mjs.v33i2.1091.

[14] Al-Tameemi, A.S. (2021). Exact and local search algorithms to minimize multicriteria scheduling problem. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10): 7821-7831. https://doi.org/10.17762/turcomat.v12i10.5772

[15] Arık, O.A. (2021). Single machine earliness/tardiness scheduling problem with grey processing times and the grey common due date. Grey Systems: Theory and Application, 11(1): 95-109. https://doi.org/10.1108/GS-01-2020-0010

[16] Hameed, A.S., Chachan, H.A. (2020). Genetic algorithm and particle swarm optimization techniques for solving multi-objectives on single machine scheduling problem. Ibn AL-Haitham Journal for Pure and Applied Science, 33(1): 119. https://doi.org/10.30526/33.1.2378

[17] Chachan, H.A., Hameed, A.S. (2019). Exact methods for solving multi-objective problem on single machine scheduling. Iraqi Journal of Science, 60(8): 1802-1813. https://doi.org/10.24996/ijs.2019.60.8.17

[18] Chachan, H.A., Jaafar, H.A. (2020). Exact solutions for minimizing cost function with five criteria and release dates on single machine. Ibn AL-Haitham Journal for Pure and Applied Science, 33(3): 140-157. https://doi.org/10.30526/33.3.2479

[19] Kalaf, B.A., Mohammed, G.J., Salman, M.D. (2021). A new hybrid meta-heuristics algorithm to solve APP problems. In Journal of Physics: Conference Series, 1897(1): 012011. https://doi.org/10.1088/1742-6596/1897/1/012011

[20] Hassan, D.A., Amiri, N.M., Ramadan, A.M. (2022). A heuristic approach to minimize three criteria using efficient solutions. Indonesian Journal of Electrical Engineering and Computer Science, 26(1): 334-341. https://doi.org/10.11591/ijeecs.v26.i1.pp334-341

[21] Neamah, N.M., Kalaf, B.A. (2023). Solving the multi-criteria: Total completion time, total late work, and maximum earliness problem. Periodicals of Engineering and Natural Sciences, 11(3): 46-57. https://doi.org/10.21533/pen.v11i3.3559.g1288

[22] Abdul-Razaq, T.S., Akram, A.O. (2018). Local search algorithms for multi-criteria single machine scheduling problem. Ibn AL-Haitham Journal for Pure and Applied Science, 436-451. https://doi.org/10.30526/2017.ihsciconf.1817

[23] Neamah, N.M., Kalaf, B.A. (2024). Minimizing total completion time, total earliness time, and maximum tardiness for a single machine scheduling problem. Ibn Al-Haitham Journal for Pure and Applied Sciences, 37(1): 386-402. https://doi.org/10.30526/37.1.3094

[24] Ali, F.H., Ahmed, M.G. (2020). Optimal and near optimal solutions for multi objective function on a single machine. In 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, pp. 152-156. https://doi.org/10.1109/CSASE48920.2020.9142053

[25] Ahmed, M., Ali, F. (2020). Efficient algorithms to solve tricriteria machine scheduling problem. Journal of Al-Rafidain University College for Sciences, 1: 485-493.