

Minimize Total Completion Time, Total Number of Tardy Jobs, and the Range of Lateness for Single Machine Scheduling problems

Nagham M. Neamah ^{a*}, Tahani Jabbar Khraibet ^b

a Mathematics Department, College of Science for Women, University of Baghdad, Baghdad, Iraq

b Thi-Qar Directorates of Education, Ministry of Education, Iraq

PAPER INFO

Received: 21.10.2025

Accepted: 09.12.2025

Published: 31.03.2026

Keywords:

Completion Time, Number of Tardy Jobs, Range of Lateness Time, Single Machine, Scheduling, Dominance Rules.



Abstract

Machine scheduling challenges are growing more intricate and dynamic. The scale and complexity of these problems demand the creation of approaches and solutions that can deliver satisfactory outcomes within an acceptable timeframe. In response, this paper introduces a novel mathematical framework tailored to multi-objective functions, focusing on single-machine scheduling issues by minimizing the discounted total completion time ($\sum C_j$), the total number of tardy jobs ($\sum U_j$), and the range of lateness time (R_L) denoted $1/(\sum C_j, \sum U_j, R_L)$ or CUR-TC. Also, minimizing the sum of total completion time ($\sum C_j$), the total number of tardy jobs ($\sum U_j$), and the range of lateness time (R_L) denoted $1/(\sum C_j + \sum U_j + R_L)$ or CUR-TO. In this paper, we prove the existence of an efficient sequence for problem $1/(\sum C_j, \sum U_j, R_L)$, where jobs are sorted in ascending order based on their processing times. Certain specific cases are highlighted to provide efficient and, in some instances, optimal solutions to the problem under consideration. The dominance rule can be used to solve problems to improve them and find efficient solutions. Accordingly, we proposed two heuristic methods for solving search problems. These are two fast methods for finding (near-optimal) solutions to optimization problems with the least possible number of tables, in comparison to the complete enumeration method and the branch and bound method. The first method is based on dominance rules, while the second method is based on MST and SPT rules.

DOI: 10.53851/psijk.v3.i9.52-61

1. INTRODUCTION

Scheduling is a popular topic in operations (OR) and production management. A variety of research industries, including manufacturing facilities, are affected by the machine scheduling problem (MSP), which is addressed through scheduling theory. It involves determining how resources will be used over time to accomplish a set of tasks and is defined as the challenge of scheduling n jobs, each requiring one or more operations, on one or more machines within specific time frame to minimize a particular objective function (Nagham M. Neamah and Bayda A. Kalaf, 2024a). The structural nature of task ordering, especially as the number of tasks and their constraints increase, is the root cause of the complexity of scheduling problems. Since 1954, scheduling issues have gotten a lot of

attention in the literature. Initially, the researchers focused on only one objective function. In practice, the decision-maker must choose only one of several objectives. Multi-criteria scheduling problems are now receiving more attention in studies (Nagham M. Neamah and Bayda A. Kalaf, 2023). Generally speaking, scheduling entails allocating machines to tasks so that all of the tasks may be finished within the given limitations (Nagham M. Neamah and Bayda A. Kalaf, 2024b). The challenge is to determine which sequence these jobs should be processed on each machine in order to reduce the specified goal function (Nagham M. Neamah & Bayda A. Kalaf, 2024). The best and most efficient way to minimize a multi-objective function is to use a table, which is a multi-objective function solver (Hoogeveen, 1996). Instead of producing a single optimal solution,

*Corresponding Author Institutional Email:

naghamm_math@cs.w.uobaghdad.edu. (Nagham M. Neamah)

these objective functions produce multiple solutions (Khraibet & Ghafil, 2021). Each objective function has multiple optimal solutions, making it impossible to choose one solution over another. According to Hooegeven (1996), a table, which is the solution to the MSP, is the most efficient method for minimizing a multiple objective function. These objective functions produce multiple optimal solutions rather than just one. Each objective function has more than one optimal solution, and therefore it is impossible to choose one that is better than two.] Non-dominated solutions are the best options in this case (Khraibet et al., 2025). Multi-objective scheduling challenges on a single machine are highly fascinating because they mimic real-world situations in which decision makers must balance multiple performance criteria to achieve the best results (Neufeld et al., 2023). It also adds several objectives to the classic issues, objectives that are often incompatible. This is an important area of research because it enhances operational efficiency and meets the needs of multiple stakeholders (Ezugwu, 2024). In most scheduling theory research, it is assumed that processing orders takes the same amount of time throughout the planning horizon. The difficulty of dealing with many objectives makes it difficult to apply classic optimization methods to multi-objective single-machine scheduling problems. However, they can be excellent tools for tackling well-defined scheduling problems. The branch and bound (BAB) algorithm was introduced by Tapan Sen et al. (1988) (Tapan S., Farhad M. E., 1988) to solve the $1//\sum C_j + R_L$ problem. By considering a linear combination of the two objectives, they were able to solve the problem with $n \leq 15$. The algorithm developed by Liao and Huang (1991) (Liao & Huang, 1991) is designed to minimize the range of delay (R_L) on a single device. Addressing the complexity of multi-objective scheduling problems often requires employing advanced multi-objective optimization methods, metaheuristic In this study, we algorithms, or hybrid strategies [6]. focused on the theoretical element by obtaining the best solution in a direct way without the need for precise methodologies by proving several special cases.

The rest of this article is organized as follows: In Sections 2 and 4, we propose two mathematical models that aim to address the three conflicting criteria in the single-device scheduling problem. Sections 3 and 5 present specific cases that demonstrate the identification of effective (optimal) solutions for the given problem. The dominance rule is described in Section 6. Section 7 presents and discusses the main outcomes from the preceding section. Conclusions are given in Section 8.

Prior to presenting the mathematical model, the following notations were used in the creation of the multi-objective model for MSP:

- $g = (g(1), g(2), \dots, g(m))$
- Av: Average.
- ANS: Average number of efficient solutions.
- AT/S: Average of CPU-Time per second
- CUR-TC: Mathematical Formulation of Problem $1//(\sum C_j, \sum U_j, R_L)$,
- CUR-TO: Mathematical Formulation of Problem $1//\sum C_j + \sum U_j + R_L$,
- $C_{g(j)}$: Duration necessary to finish job j within sequence g . [Time required to complete task (job) j in sequence g]
- $\sum C_{g(j)}$: Total completion time within sequence g
- $d_{g(j)}$: Due date of job j within sequence g
- EFS: Efficient solution
- $E_{g(j)}$: Earliness of job j within sequence g
- $E_{max(g)}$: Maximum earliness inside sequence g
- F_{CUR-TC} : Objective Function of $1//(\sum C_j, \sum U_j, R_L)$,
- F_{CUR-TO} : Objective Function of $1//\sum C_j + \sum U_j + R_L$,
- $L_{g(j)}$: The lateness of job j within sequence g
- $L_{max(g)}$: Maximum lateness within sequence g
- $L_{min(g)}$: Minimum lateness within sequence g
- $M = \{1, 2, \dots, m\}$: A collection of tasks requiring scheduling.
- OPS: Optimal solution
- $p_{g(j)}$: Processing time of job j within sequence g
- $R: 0 < Real < 1$
- $R_{L(g)}$: The Range of Lateness within sequence g
- $U_{g(j)}$: Number of tardy jobs within sequence g
- $\sum U_{g(j)}$: Total number of tardy times within the sequence g
- TCMSP: Tri-Criteria Machine Scheduling Problem
- TOMSP: Tri-Objective Machine Scheduling Problem

Definition (1): When a schedule g is unable to meet $f_{g(j)} \leq f_{g(j)}, j = 1, \dots, k$, with at least one of the aforementioned holdings as a rigorous inequality, then

that schedule is considered efficient. This is also known as g' dominating g .

Definition (2): Let $(h_{i(0)}: i = 1,2,3)$ be a solution for the tri-criteria problem $1//(\sum C_j, \sum U_j, R_L)$, then the Euclidean distance (d) for this solution is: $d = \sqrt{\sum h_{i(0)}^2} =$

$$\sqrt{h_{1(0)}^2 + h_{2(0)}^2 + h_{3(0)}^2} \quad (1)$$

Note (1): A useful measure for determining the most efficient solution from a Pareto-optimal set is the distance d .

Proposition (1): Let $(h_{1(0)}, h_{2(0)}, h_{3(0)})$ be a solution for the tri-criteria problem $1//(\sum C_j, \sum U_j, R_L)$, and $h_{i(0)} \neq 0, i = 1,2,3$, then always: $h_{i(0)} \leq d, i = 1,2,3$.

Proof: Let's assume that $h_{i(0)} > d$, from (1):

$$h_{i(0)} > \sqrt{h_{1(0)}^2 + h_{2(0)}^2 + h_{3(0)}^2}$$

$$h_{1(0)}^2 > h_{1(0)}^2 + h_{2(0)}^2 + h_{3(0)}^2$$

This is a contradiction since the above inequality is not true even $h_{i(0)} = 0, i = 1,2,3$.

2. Problem Description

After reviewing the terminology of the mathematical model, we move on to constructing the mathematical model of the research topic: Let $M = \{1,2, \dots, m\}$ be a set of jobs that want to be scheduled on a tri-criteria machine scheduling problem (TCMSP) with $p_j \leq d_j$, and the tri-criteria machine scheduling problem (TCMSP) for each NP-hard problem. The MSP can only process one job at a time using the three fields' classification. Using the table $g = (g(1), g(2), \dots, g(m))$, the discussed TCMSP is denoted by $1//(\sum C_j, \sum U_j, R_L)$, and CUR-TC. which can be formulated as follows:

$$F_{CUR-TC} = \text{Min} \left\{ \begin{array}{l} \sum C_{g(j)} \\ \sum U_{g(j)} \\ R_{L(g)} \end{array} \right. \quad (2)$$

$$\left. \begin{array}{l} \text{Subject to} \\ C_0 = 0 \\ C_{g(j)} = C_{g(j-1)} + p_{g(j)} \\ C_{g(j)} > 0 \end{array} \right\} \quad (3)$$

$$\left. \begin{array}{l} L_{g(j)} = C_{g(j)} - d_{g(j)} \\ L_{max(g)} = \max_{1 \leq j \leq m} \{L_{g(j)}\} \\ L_{min(g)} = \min_{1 \leq j \leq m} \{L_{g(j)}\} \end{array} \right\} \quad (4)$$

$$\left. \begin{array}{l} U_{g(j)} = 0 \quad \text{if } L_{g(j)} \leq 0 \\ U_{g(j)} = 1 \quad \text{if } L_{g(j)} > 0 \end{array} \right\} \quad (5)$$

$$\left. \begin{array}{l} E_{g(j)} \geq d_{g(j)} - C_j \\ E_{max(g)} = \max_{1 \leq j \leq m} \{E_{g(j)}\} \\ E_{g(j)} \geq 0 \end{array} \right\} \quad (6)$$

$$\left. \begin{array}{l} R_{L(g)} = L_{max(g)} + E_{max(g)} \\ E_{max(g)} = -L_{min(g)} \\ R_{L(g)} \geq 0 \end{array} \right\} \quad (7)$$

In this paper, we will try to find an efficient solution to CUR-TO, given the difficulty of solving this problem.

Proposition (2): There is an efficient sequence for problem $1//(\sum C_j, \sum U_j, R_L)$, which obeys the shortest processing time (SPT) rule.

Proof: (i) Firstly, assume that $p_i \neq p_j$ for all i, j . The unique sequence SPT_1 , and consequently (SPT_2) , delivers the absolute minimum value of $\sum C_j$. Thus, there is no alternative sequence $\delta \neq SPT_2$ such that.

$$\left. \begin{array}{l} \sum C_{j(\delta)} \leq \sum C_{j(SPT_2)}, \\ \sum U_{j(\delta)} \leq \sum U_{j(SPT_2)}, \\ \text{and } R_{L(\delta)} \leq R_{L(SPT_2)} \end{array} \right\} \quad (8)$$

At least one strict inequality must be present.

(ii) If multiple SPT_1 sequences exist for certain jobs with identical processing times, the SPT_2 sequence should adhere to the SPT_1 rule while prioritizing jobs with the same processing time in EDD order (noting that EDD and MOOR sequences are equivalent). For a subset of jobs categorized as early or relatively early, this EDD = MOOR sequence effectively minimizes $L_{max}, \sum U_j$.

The due date is the same for all jobs that take the same amount of time to process. If there are periods when things are slow, then SPT_2 is not unusual. Demonstrate that any SPT_2 sequencing constitutes an efficient sequencing that does not adhere to the SPT_1 rule, which states that an SPT_1 cannot dominate an SPT_2 sequencing, as indicated in (8). If δ is an SPT sequence, it cannot be SPT_2 sequencing since it cannot dominate SPT_2

$$\left. \begin{array}{l} \sum C_{j(\delta)} = \sum C_{j(SPT_2)}, \\ \sum U_{j(SPT_2)} \leq \sum U_{j(\delta)}, \\ \text{and } R_{L(SPT_2)} \leq R_{L(\delta)} \end{array} \right\} \quad (9)$$

Due to the implementation of the EDD rule, all of the SPT_2 sequences function effectively.

3. SPECIAL CASES FOR PROBLEM CUR-TC

Case (3.1): For problem CUR-TC, if the processing times for all jobs are uniform and the due dates are similarly uniform ($p_j = p$ and $d_j = d, \forall j$), Consequently, a singular optimal solution exists with $n!$ sequences under a constant objective function.

Proof: As known:

$$\begin{aligned} C_{g(j)} &= \sum_{k=1}^j p_k = jp, \text{ then } \sum_{j=1}^m C_{g(j)} = p \frac{m(m+1)}{2}, \text{ and} \\ L_{g(j)} &= C_{g(j)} - d_{g(j)} = jp - d \\ L_{max(g)} &= \max_{1 \leq j \leq m} \{L_{g(j)}\} = mp - d \quad \text{and} \quad L_{min(g)} = \\ &= \min_{1 \leq j \leq m} \{L_{g(j)}\} = p - d \\ \therefore R_{L(g)} &= L_{max(g)} - L_{min(g)} = mp - d - (p - d) = \\ &= mp - p = (m - 1)p \end{aligned} \quad (10)$$

$$E_{g(j)} = \max\{d_{g(j)} - C_{g(j)}, 0\} = \max\{d - jp, 0\}$$

$$E_{g(j)} = \max\{d - p, d - 2p, \dots, d - jp, \dots, d - mp, 0\}$$
(11)

$$T_{g(j)} = \max\{C_{g(j)} - d_{g(j)}, 0\} = \max\{jp - d, 0\}$$

$$T_{g(j)} = \max\{p - d, 2p - d, \dots, jp - d, \dots, mp - d, 0\}$$
(12)

1) If all jobs are early, this means $d > C_{g(j)} = jp$ & $L_{g(j)} \leq 0$, then $U_{g(j)} = 0$ and the maximum difference in relation (11) is $d - p$, then:

a) If $L_{g(j)} = 0, \forall j$, hence $L_{max(g)} = L_{min(g)} = 0$. Then

$$(\sum C_{g(j)}, \sum U_{g(j)}, R_L) = (p \frac{m(m+1)}{2}, 0, 0) \quad (13)$$

b) If $L_{max(g)} = 0$ and $L_{min(g)} < 0, E_{max(g)} = -L_{min(g)} = d - p$, then

$$(\sum C_{g(j)}, \sum U_{g(j)}, R_L) = (p \frac{m(m+1)}{2}, 0, d - p) \quad (14)$$

c) If $L_{max(g)} = C_{max(g)} - d < 0$ and $L_{min(g)} = p_{g(1)} - d < 0$, then:

$$R_L(g) = C_{max(g)} - p_{g(1)}, \text{ and}$$

$$(\sum C_{g(j)}, \sum U_{g(j)}, R_L) = (p \frac{m(m+1)}{2}, 0, C_{max(g)} - p_{g(1)}) \quad (15)$$

From (13), (14), and (15), we obtain that there exists a unique efficient solution with $n!$ sequences with a constant objective function

2) If all jobs are late (except the 1st job, this means $d = p$), $U_{g(j)} = 1$, and $E_{g(j)} = 0, \forall j$, then:

$$E_{max(g)} = 0 \quad (16)$$

$$L_{max(g)} = \max_{1 \leq j \leq m} \{L_{g(j)}\} = mp - d = mp - p = (m - 1)p \text{ and } L_{min(g)} = p - d = 0$$

$$R_L(g) = L_{max(g)} - L_{min(g)} = (m - 1)p - 0 = (m - 1)p \quad (17)$$

From relations (15) and (16), we obtain:

$$(\sum C_{g(j)}, \sum U_{g(j)}, R_L) = (p \frac{m(m+1)}{2}, m - 1, (m - 1)p).$$

3) If $p_{g(1)} = C_{g(1)} < d < C_{g(j>1)}$:

$$L_{max(g)} = \max_{1 < j \leq m} \{L_{g(j)}\} = \max_{1 < j \leq m} \{C_{g(j)} - d, 0\}.$$

$$L_{max(g)} = C_{max(g)} - d \quad (18)$$

Then:

$$T_{max} = C_{max(g)} - d \quad (19)$$

$$L_{min(g)} = \min_{1 \leq j \leq m} \{L_j\} = -E_{max(g)} = -(d - p) \quad (20)$$

From relations (18) and (20), $R_L(g) = T_{max} - (-E_{max(g)}) = T_{max} + E_{max(g)} = C_{max(g)} - d + (d - p) = mp - p = (m - 1)p$.

From relations (18), (19), and (20), the CUR-TCMSP is converted to $(\sum C_{g(j)}, \sum U_{g(j)}, R_L) = (p \frac{m(m+1)}{2}, k, (m - 1)p)$, where k represents the number of delayed jobs (k is constant).

Case (3.2): When the due date for all tasks is uniform (denoted by $d_j = d$, for each j), applying the rule $g =$

SPT provides an efficient solution to the CUR-TC problem.

Proof: Let's have the sequence $g = SPT, L_{max(g)} = C_{max(g)} - d$ and $L_{min(g)} = C_{min(g)} - d = p_{g(1)} - d$.

$$R_L(g) = L_{max(g)} - L_{min(g)} = \max_{1 \leq j \leq m} \{L_{g(j)}\} - \min_{1 \leq j \leq m} \{L_{g(j)}\} = \max_{1 \leq j \leq m} \{C_{g(j)} - d\} - \min_{1 \leq j \leq m} \{C_{g(j)} - d\} =$$

$$\max_{1 \leq j \leq m} \{C_{g(j)}\} - d - \left\{ \min_{1 \leq j \leq m} \{C_{g(j)}\} - d \right\} = \max_{1 \leq j \leq m} \{C_{g(j)}\} - d - \min_{1 \leq j \leq m} \{C_{g(j)}\} + d = \max_{1 \leq j \leq m} \{C_{g(j)}\} -$$

$$\min_{1 \leq j \leq m} \{C_{g(j)}\} = \max_{1 \leq j \leq m} \{C_{g(j)}\} - p_{g(1)} = C_{max(g)} -$$

$$p_{g(1)} = \sum_{j=2}^m p_{g(j)}.$$

1) If $d \geq C_{g(j)}$:

$T_{max(g)} = 0$ (Since all jobs are early), $U_{g(j)} = 0$ then $1//(\sum C_{g(j)}, \sum U_{g(j)}, R_L(g))$ reduce to $1//(\sum C_{g(j)}, R_L(g)) = (\sum C_{g(j)}, C_{max(g)} - p_{g(1)}) = (\sum C_{g(j)}, \sum_{j=2}^m p_{g(j)})$. In this case, the SPT rule gives an efficient solution to CUR-TCMSP.

2) If $d < C_{g(j)}, \forall j$:

$L_{max(g)} = \max_{1 \leq j \leq m} \{L_{g(j)}\} = \max_{1 \leq j \leq m} \{C_{g(j)} - d, 0\} = C_{max(g)} - d = T_{max(g)}, L_{min} = \min_{1 \leq j \leq m} \{L_{g(j)}\} = -E_{max(g)} = 0$ ($E_{max(g)} = 0$ Since all jobs are late), $R_L(g) = C_{max(g)} - d$, and $\sum_{j=1}^m U_{g(j)} = m$. Then $1//(\sum C_{g(j)}, \sum U_{g(j)}, R_L(g)) = 1//(\sum C_{g(j)}, m, C_{max(g)} - d)$. In this case, the SPT rule gives an efficient solution to CUR-TCMSP.

Case (3.3): For problem CUR-TC,

- 1) For $g = SPT$, if all jobs are neither early and not late ($C_{g(j)} = d_{g(j)}, \forall j$), then we obtain an efficient solution.
- 2) For $g = EDD$, if all jobs are early ($C_{g(j)} < d_{g(j)}, \forall j$), then we obtain an efficient solution.
- 3) For $g = MST$, if all jobs are late ($C_{g(j)} \geq d_{g(j)}, \forall j$), then we obtain an efficient solution.

Proof: 1. Since $(C_{g(j)} = d_{g(j)}, \forall j)$, then $T_{max(g)} = E_{max(g)} = 0$. Hence, $1//(\sum C_{g(j)}, \sum U_{g(j)}, R_L(g))$ reduces to $1//(\sum C_{g(j)})$; this problem has an optimal solution and an efficient solution for CUR-TC when using $g = SPT$. $1//(\sum C_{g(j)}, \sum U_{g(j)}, R_L(g)) = 1//(\sum C_{g(j)})$.

2. Since all jobs are early for $g = MST$, thus: $T_{max(g)} = \sum U_{g(j)} = 0$, then $L_{max(g)} \leq 0$ and $L_{min(g)} \leq 0$. Notice that $L_{max(g)}$ is the minimum L_{max} and $L_{min(g)}$ is the minimum L_{min} for all feasible sequences, then $R_L(g) = L_{max(g)} - L_{min(g)}$ is the minimum R_L for all feasible sequences. Since $C_{g(j)} \leq d_{g(j)}, \forall j$, with minimum $R_L(g)$ and $T_{max(g)} = \sum U_{g(j)} = 0$, then g gives an efficient solution for CUR-TCMSP.

$1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)}) = 1/(\sum C_{g(j)}, 0, R_{L(g)}) = 1/(\sum C_{g(j)}, L_{max(g)} - L_{min(g)})$
 3. Since $(C_{g(j)} \geq d_{g(j)}, \forall j)$, thus: $E_{max(g)} = 0$, then $1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)}) = 1/(\sum C_{g(j)}, m, R_{L(g)}) = 1/(\sum C_{g(j)}, m, L_{max(g)})$, then g gives an efficient solution for CUR-TC.

Case (3.4): For problem CUR-TC: If $p = p_j, \forall j$, then the CUR-TC can be solved by using the EDD rule.

Proof: Let's have the sequence $g = EDD$.

As known $L_{g(j)} = C_{g(j)} - d_{g(j)}$, and since $p = p_j, \forall j$ then $C_j = jp, \sum_{j=1}^m C_j = \sum_{j=1}^m jp = p \frac{m(m+1)}{2}$ (21)

So, $L_{g(j)} = jp - d_{g(j)} = \{p - d_{g(1)}, 2p - d_{g(2)}, \dots, np - d_{g(m)}\}$
 $R_{L(g)} = L_{max(g)} - L_{min(g)} = \max_{1 \leq j \leq m} \{L_{g(j)}\} - \min_{1 \leq j \leq m} \{L_{g(j)}\} = \max_{1 \leq j \leq m} \{jp - d_{g(j)}\} - \min_{1 \leq j \leq m} \{jp - d_{g(j)}\} = mp - d_{max(g)} - (p - d_{min(g)})$
 $R_{L(g)} = (m - 1)p - (d_{max(g)} - d_{min(g)})$ (22)

1) If $d_{max(g)} < C_m = C_{max(g)} = mp$, then $L_{max(g)} = mp - d_{max(g)} > 0 \therefore T_{L_{max(g)}} = L_{max(g)} = mp - d_{max(g)} = R_{L(g)}$, & $\sum_{j=1}^m U_{g(j)} = m$ (23)

2) If $C_m \leq d_{min(g)}$, then all jobs are early $\therefore T_{L_{max(g)}} = 0$ & $\sum_{j=1}^m U_{g(j)} = 0$ (24)

3) $C_{g(j)} = jp = d_{g(j)}$, this means $T_{max(g)} = 0$ & $E_{max(g)} = 0$, then $\sigma = EDD$, and $L_{max(g)} = L_{min(g)} = 0$, then $R_{L(g)} = 0$ & $\sum_{j=1}^m U_{g(j)} = 0$.

From relations (21), (22), (23), and (24), If all functions are constants, then any sequence has a job order with $d_{min(g)}$ and $d_{max(g)}$ at the first and last positions, respectively, provides the same unique efficient solution, and there are $(n - 2)!$ sequences. This relation depends on the variable. $d_{g(j)}$, then $R_{L(g)}, U_{g(j)}$ depend on $d_{g(j)}$ only, then the EDD rule gives an optimal solution for CUR-TC.

Case (3.5): For problem CUR-TC, if the rules for MOOR, EDD, and SPT are the same, we have an effective solution.

1) If $g = SPT = EDD = MST = MOOR$ at the same time, then there is an efficient solution.

2) If $g = SPT = EDD = MOOR$ and p_j and d_j are different, then there is only one efficient solution.

Proof:

1) It is well known that the SPT rule minimizes $\sum C_{g(j)}$, the EDD rule minimises $T_{max(g)}$, and the MOOR rule minimizes $\sum U_{g(j)}$. The ideal solution for $\sum C_{g(j)}, T_{max(g)}$, and $\sum U_{g(j)}$ will be provided by a schedule g s.t $g = SPT = EDD = MOOR$ simultaneously.

2) Similar to (1), if we have $g = SPT = EDD = MOOR$ and all p_j and d_j are distinct from one another, then g

will be unique and there won't be any τ s.t. $\sum C_{\tau(j)} \leq \sum C_{g(j)}, \sum U_{\tau(j)} \leq \sum U_{g(j)}$, and $R_{L(\tau)} \leq R_{L(g)}$, where at least one of the inequalities is stringent. Then, regardless of whether σ minimises R_L or not, it provides a special, effective solution for CUR-TC in two scenarios.

4. Mathematical Formulation of Subproblem for CUR-TOMSP

In this section, we will discuss the sub-problem of CUR-TC; we will try to find the optimal solution using the table $g = (g(1), g(2), \dots, g(m))$. This is a tri-objective problem (TOMSP), which is defined as $1/(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)})$, and is denoted by CUR-TO. It can be formulated as follows:

$$F_{CUR-TO} = \text{Min}(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) \quad (25)$$

Subject to (3) - (7)

Proposition (3): For problem CUR-TO, and schedule g . If $m \rightarrow \infty$, then $1/(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) \rightarrow \infty$.

Proof: Since the range of lateness ($R_{L(g)}$) is equal to the difference between the maximum lateness time ($L_{max(g)} = \max_{1 \leq j \leq m} \{L_{g(j)}\}$), the minimum lateness time

($L_{min(g)} = \min_{1 \leq j \leq m} \{L_{g(j)}\}$) if $m \rightarrow \infty$, $\max_{1 \leq j \leq m} \{L_{g(j)}\} \rightarrow \infty$, since $\max_{1 \leq j \leq m} \{L_{g(j)}\} \rightarrow L_m$ and since $C_{g(j)} = C_{max} \gg d_m$, then

$\therefore L_m \rightarrow \infty$, since $R_{L(g)} = \max_{1 \leq j \leq m} \{L_{g(j)}\} - \min_{1 \leq j \leq m} \{L_{g(j)}\}$, that implies $\lim_{m \rightarrow \infty} R_{L(g)} = L_m - L_{min(g)} \approx L_m$ as $m \rightarrow \infty$.

$\therefore R_{L(g)} \rightarrow \infty$
 CUR-TOMSP, if $m \rightarrow \infty$, then $\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}$.

The Proposition below describes the relationship between CUR-TC and CUR-TO and shows that any optimal solution to problem CUR-TO is an efficient solution to problem CUR-TC.

Proposition (4): Let g represent the schedule that provides one of the efficient solutions for CUR-TC, if and only if g represents the optimal solution for CUR-TO.

Proof: Let $Q = \{\mu_1, \mu_2, \dots, \mu_r\}$ represent the set of all efficient schedules yielding efficient outcomes $((g, h, w) = (g_i, h_i, w_i)) \forall i = 1, 2, \dots, r$, in the framework of CUR-TC. Let us consider that the optimal solution $(g_k + h_k + w_k)$ for CUR-TOMSP does not belong to the set of efficient solutions for CUR-TC. Under this assumption, the schedule g , which yields $(g_k + h_k + w_k)$ such that $g \notin Q$, leads to the conclusion that $g_i + h_i + w_i < g_k + h_k + w_k, \forall i = 1, 2, \dots, r$. This implies that $g_k + h_k + w_k$ cannot be an optimal solution for CUR-TO, which creates a logical contradiction.

Now reconsider g as a schedule providing the optimal solution $g_k + h_k + w_k$ for CUR-TOMSP. Suppose g does not deliver an efficient solution for CUR-TC; this condition implies that $g \notin S$ meaning there exists an efficient solution $g_i + h_i + w_i < g_k + h_k + w_k$. Consequently, $(g_i + h_i + w_i)$ would serve as the optimal solution for CUR-TO and achieve a value smaller than $g_k + h_k + w_k$. This reasoning again leads to a contradiction since $g_k + h_k + w_k$ cannot simultaneously be a valid optimal solution for CUR-TO under these circumstances. Hence, the initial assumption must be incorrect.

5. The key special cases of the CUR-TO

Case (5.1): For problem CUR-TO, if the processing times of all jobs are identical and the due date is also identical ($p_j = p$ and $d_j = d, \forall j$), then there exists a unique optimal solution with $n!$ sequences with a constant objective function

Proof: From case (3.1) for problem CUR-TO, we prove that:

1) If all jobs are early, this means $d > C_{g(j)} = jp$, then $(\sum C_{g(j)}, \sum U_{g(j)}, R_L) = (p \frac{m(m+1)}{2}, 0, (m-1)p)$

So, for CUR-TO, we obtain: $(\sum C_{g(j)} + \sum U_{g(j)} + R_L) = (p \frac{m(m+1)}{2} + 0 + (m-1)p) = p \frac{m^2+3m-2}{2}$

2) If all jobs are late (except the 1st job), then $(\sum C_{g(j)}, \sum U_{g(j)}, R_L) = (p \frac{m(m+1)}{2}, m-1, (m-1)p)$

So, for CUR-TO, we obtain: $(\sum C_{g(j)} + \sum U_{g(j)} + R_L) = (p \frac{m(m+1)}{2} + m-1 + (m-1)p) = p \frac{m^2+3m-2}{2} + m-1$.

3) If $p_{g(1)} = C_{g(1)} < d < C_{g(j>1)}$, then we prove $(\sum C_{g(j)}, \sum U_{g(j)}, R_L) = (p \frac{m(m+1)}{2}, k, (m-1)p)$,

where k represents the number of delayed jobs (k is constant).

So, for CUR-TO, we obtain: $(\sum C_{g(j)} + \sum U_{g(j)} + R_L) = p \frac{m(m+1)}{2} + k + (m-1)p = p \frac{m^2+3m-2}{2} + k$

Case (5.2): When the due date for all tasks is uniform (denoted by $d_j = d$, for each j), applying the rule $g = SPT$ provides an efficient solution to the CUR-TO problem.

Proof: From case (3.2) of CUR-TC:

1. If $d \geq C_{g(j)}$; we prove that: $1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)})$ reduce to $1/(\sum C_{g(j)}, R_{L(g)}) = (\sum C_{g(j)}, C_{max(g)} - p_{g(1)}) = (\sum C_{g(j)}, \sum_{j=2}^m p_{g(j)})$.

So, for CUR-TO, we obtain: $(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) = (\sum C_{g(j)} + R_{L(g)}) = (\sum C_{g(j)} + C_{max(g)} - p_{g(1)}) = (\sum C_{g(j)} + \sum_{j=2}^m p_{g(j)})$

2. If $d < C_{g(j)}, \forall j$; we prove that: $1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)}) = 1/(\sum C_{g(j)}, m, C_{max(g)} - d)$.

So, for CUR-TO we obtain: $1/(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) = 1/(\sum C_{g(j)} + m + C_{max(g)} - d)$.

Case (5.3): For problem CUR,

1) For $g = SPT$, if all jobs are neither early and not late ($C_{g(j)} = d_{g(j)}, \forall j$), then we obtain an optimal solution.

2) For $g = MST$, if all jobs are early ($C_{g(j)} \leq d_{g(j)}, \forall j$), then we obtain an optimal solution.

3) For $g = EDD$, if all jobs are late ($C_{g(j)} \geq d_{g(j)}, \forall j$), then we obtain an optimal solution.

Proof: From case (3.3) of CUR-TC:

1) If $(C_{g(j)} = d_{g(j)}, \forall j)$; we prove that: $1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)}) = 1/(\sum C_{g(j)})$

So, for TOMSP-CUR, we obtain: $1/(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) = 1/(\sum C_{g(j)})$

2) If $C_{g(j)} \leq d_{g(j)}, \forall j$; we prove that: $1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)}) = 1/(\sum C_{g(j)}, 0, R_{L(g)})$

So, for CUR-TO, we obtain: $(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) = \sum C_{g(j)} + 0 + R_{L(g)}$

3. If $(C_{g(j)} \geq d_{g(j)}, \forall j)$; we prove that: $1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)}) = 1/(\sum C_{g(j)}, m, R_{L(g)}) = 1/(\sum C_{g(j)}, m, L_{max(g)})$

So, for CUR-TO, we obtain: $(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) = \sum C_{g(j)} + m + R_{L(g)} = \sum C_{g(j)} + m + L_{max(g)}$.

Case (5.4): For problem CUR-TO-TOMSP if $p = p_j, \forall j$, then the problem CUR can be solved by the EDD rule.

Proof: (See case (3.4) of CUR-TC). [Refer to case (3.4) of CUR-TC]

1) If $d_{max(g)} < C_m = C_{max(g)} = mp$, we prove $1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)}) = 1/(\sum C_{g(j)}, m, mp - d_{max(g)})$. So, we obtain: $1/(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) = 1/p \frac{m(m+1)}{2} + m + mp - d_{max(g)}$.

2) If $C_m \leq d_{min(g)}$, we prove $1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)}) = 1/(\sum C_{g(j)}, R_{L(g)})$. So, we obtain: $1/(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) = 1/(\sum C_{g(j)} + R_{L(g)}) = p \frac{m(m+1)}{2} + mp - p$.

3) If $C_{g(j)} = jp = d_{g(j)}$, we prove $1/(\sum C_{g(j)}, \sum U_{g(j)}, R_{L(g)}) = 1/(\sum C_{g(j)})$. So, we obtain: $1/(\sum C_{g(j)} + \sum U_{g(j)} + R_{L(g)}) = 1/(\sum C_{g(j)}) = p \frac{m(m+1)}{2}$.

Case (5.5): For problem CUR-TO, if SPT, EDD, and MOOR rules are identical, then we obtain an efficient solution.

- 1) If $g = SPT = EDD = MST = MOOR$ at the same time, then \exists an efficient solution.
- 2) If p_j and d_j are all different and $g = SPT = EDD = MOOR$, then \exists a unique efficient solution.

Proof: (see case (3.5) of CUR–TC).

	Job ₁	Job ₂	Job ₃	Job ₄
p_j	3	11	12	6
d_j	16	14	29	25
$S_j=d_j - p_j$	13	3	17	19

6.Heuristic Methods for CUR–TTCMSP and CUR–TOMSP

When solving a scheduling problem, implicit counting techniques (BAB and CEM) are typically used to determine optimal solutions. Precise techniques cannot address large problems and require a very long time to find the best answer. When it comes to an NP-hard problem, the computational requirements are enormous for a large-scale problem. To avoid these drawbacks, we can resort to heuristics. Heuristic methods (HMs) are search processes that identify (near-optimal) solutions to optimization problems in a short period of time. It can also be defined as, A heuristic approach is a methodology that aims to identify a decent solution (i.e., one that is near ideal) at a reasonable computing cost. It cannot ensure optimality or feasibility, or even assess how close a potential solution is to optimal in many circumstances. First method: DR–CUR, based on DRs, is proposed to solve the CUR–TC and CUR–TO DR–CUR is based on identifying the type of sequence with minimal p_j and s_j , which is compatible with DRs, and calculating the objective function. The DR–CUR - based method is summarized in the following steps:

Algorithm (1): DR– CUR Heuristic Method for CUR– TCMSP(CUR– TOMSP)

- Step 1:** Input: m, p_j and $d_j, j = 1, 2, \dots, m$.
- Step 2:** To determine the dominant relationships (DRs) and the corresponding adjacency matrix A ; $M = \{1, 2, \dots, m\}$; compute $s_j = d_j - p_j, \forall j \in N, \mu = \varphi$, if $p_i \leq p_j$ and $s_i \leq s_j$, an efficient solution exists where job i precedes job j .
- Step 3:** Identify a sequence α_1 that maintains a non-increasing order of p_j without conflicting with the DR (matrix A); if multiple job orders α_1 exist based on d_j , then $\mu = \mu \cup \{\alpha_1\}$.
- Step 4:** Determine a sequence α_2 that follows a non-increasing order of (arranged in non-increasing order of) s_j and does not conflict with the DR (that aligns with the DR) (matrix A); if multiple jobs disrupt links, arbitrarily arrange them (In cases where multiple jobs create link disruptions, assign an arbitrary order) α_2 based on p_j , then $\mu = \mu \cup \{\alpha_2\}$.
- Step 5:** Extract the dominant sequence μ' from μ .

Step 6: Calculate (Compute) $F_{CUR-TC}(\mu')$, and identify(obtain) $F_{CUR-TO(i)} = F_{CUR-TO}(\mu'(i)), f = \min(F_{CUR-TO(i)}, k = i)$.

Step 7: Output: Effective solution set μ' , with the optimal solution being along f with the sequence set $\mu'(k)$.

Step 8: End.

Example (1): For the CUR–TC–TCMSP, and CUR–TO–TOMSP, assume we possess the following data for = 4:

Using the following rule [If $p_i \leq p_j$ and $s_i \leq s_j$, then task i is executed before task j .], we obtain the DRs shown in Figure 1.

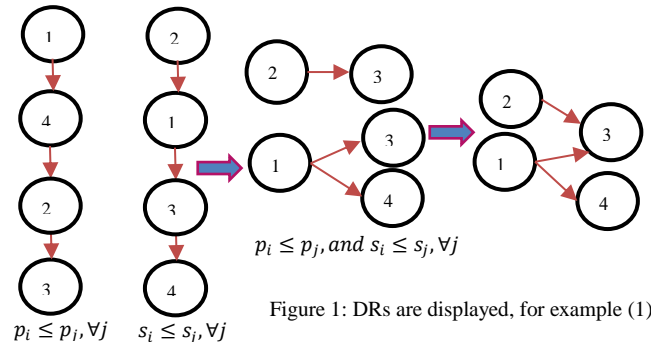


Figure 1: DRs are displayed, for example (1).

$1 \rightarrow 3, 1 \rightarrow 4$, and $2 \rightarrow 3$ are the three visible sequences. Some (or all) of the five possible sequences are controlled by the DRs described earlier in Table 1. Here is the adjacency matrix A :

$$A = \begin{bmatrix} 0 & a_{12} & 1 & 1 \\ a_{21} & 0 & 1 & a_{24} \\ 0 & 0 & 0 & a_{34} \\ 0 & a_{42} & a_{43} & 0 \end{bmatrix}$$

such that, if $i \rightarrow j$ then $a_{ij} = 1$, and $j \rightarrow i$ then $a_{ji} = 1$. The lists of possible effective sequences in example (1) subject to DR are shown in Table (1), and Table (1) shows the lists of effective sequences in example (1) subject to DR.

Table (1): lists the possible efficient sequences in Example (1) subject to DR.

Sequence	$\sum C_j$	$\sum U_j$	R_L	CUR–TC	d	CUR–TO
3,2,4,1	64	2	22	(64,2,22)	67.7052	88
2,1,4,3	77	1	8	(77,1,8)	77.4209	86
3,4,2,1	69	1	16	(69,1,16)	70.8378	86
1,2,3,4	75	1	20	(75,1,20)	77.6273	96
2,1,3,4	83	1	10	(83,1,10)	83.6062	94

Table (2): lists the efficient sequences in Example (1) subject to DR.

Sequence	$\sum C_j$	$\sum U_j$	R_L	CUR–TC	d	CUR–TO
3,2,4,1, (SPT)	64	2	22	(64,2,22)	67.7052	88
3,4,2,1	69	1	16	(69,1,16)	70.8378	86
2,1,3,4, (MST)	83	1	10	(83,1,10)	83.6062	94

Second method: The MS–CUR method is based on the following steps: First, we use the MST rule to determine the objective function. Second, we put the first job in order in the MST rule, then we arrange the remaining jobs according to the SPT rule, then we calculate the objective function. Subsequently, the second job is prioritized for scheduling by adhering to the Minimum Slack Time (MST) rule. Following this, the remaining jobs are sequentially arranged based on the Shortest Processing Time (SPT) rule. After the scheduling is established, the objective function is evaluated to assess performance outcomes. We continue in this manner until we reach a set of efficient solutions.

Algorithm (2): MS– CUR Heuristic Method for CUR– TCMSP(CUR– TOMSP)

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

Step 2: For each job j , calculate its slack time: $s_j = d_j - p_j, \forall j$, apply the MST rule (Sort the jobs in non-decreasing order of slack times).

Step 3: Create candidate sequences

For each position $k = 1, 2, \dots, n$

- Choose the first job = the k^{th} job in MST order.
- Take the remaining jobs, and sort them by the SPT (non-descending order of processing time) rule.
- Construct the filter sequence = [first job, remaining jobs].

Step 4: calculate the objective function F_{CUR-TC} , and F_{CUR-TO} for each sequence.

Step 5: Evaluate the candidate sequence

Step 6: Final output after checking all n possible first jobs: output the set of efficient solutions (optimal solution).

Step 8: End.

Example (2): For the CUR–TC–TCMSP, and CUR–TO–TOMSP, assume we possess the following data for = 4:

	Job_1	Job_2	Job_3	Job_4
p_j	10	3	10	6
d_j	20	20	25	22
$s_j = d_j - p_j$	10	17	15	16

Applying the MS–CUR method results in the following:

Step (1): The MST sequence (1,3,4,2) is the first candidate sequence with objective function $(\sum C_j, \sum U_j, R_L) = (85, 19, 2)$.

Step (2): Place the first job in order in the MST rule, then arrange the remaining jobs according to the SPT rule; we get the second filter sequence (1,2,4,3) with objective function $(\sum C_j, \sum U_j, R_L) = (71, 14, 1)$.

Step (3): Place the second job in order in the MST rule, then arrange the remaining jobs according to the SPT rule; we get the third filter sequence (3,2,4,1) with objective function $(\sum C_j, \sum U_j, R_L) = (71, 24, 1)$.

Step (4): Place the third job in order in the MST rule, then arrange the remaining jobs according to the SPT rule; we get the fourth filter sequence (4,2,1,3) with objective function $(\sum C_j, \sum U_j, R_L) = (63, 20, 1)$.

Step (5): Place the fourth job in order in the MST rule, then arrange the remaining jobs according to the SPT rule; we get the fifth filter sequence (2,4,1,3) with objective function $(\sum C_j, \sum U_j, R_L) = (60, 21, 1)$.

Step (5): Filter the solution set to get a set of efficient solutions to the problems.

Table (9) shows the lists of effective sequences in example (8).

Table (3): lists the efficient sequences in Example (2).

Sequence	$(\sum C_j, \sum U_j, R_L)$	d	$\sum C_j + \sum U_j + R_L$
(1,2,4,3)	(71,14,1)	72.3740	86
(4,2,1,3)	(63,20,1)	66.1059	84
(2,4,1,3)	(60,21,1)	63.5767	82

7. RESULTS AND DISCUSSION

In this section, the findings we have reached in the previous sections are formulated as follows:

An efficient sequence that satisfies the SPT rule exists for problem 1// $(\sum C_j, \sum U_j, R_L)$.

- The OPS for problem 1// $\sum C_j + \sum U_j + R_L$ if and only if it is one of the EFS to problem 1// $(\sum C_j, \sum U_j, R_L)$.
- If the processing time for all jobs is uniform and the due dates are also the same, there exists a unique efficient solution characterized by $n!$ sequences that maintain a constant objective function.
- If the due date of all jobs is identical, then the $g = SPT$ rule implies an EFS(OPS) solution for the problems CUR–TC (CUR–TO).
- If the processing time of all jobs is identical, then the $g = EDD$ rule implies an EFS(OPS) solution for the problems CUR–TC(CUR–TO).
- If the rules for SPT, EDD, and MOOR are identical, we can derive the EFS(OPS) solution to the CUR TC (CUR TO) problems.

In conclusion, this study presents two heuristics designed to address research problems, specifically targeting the rapid identification of (near-optimal) solutions to optimization challenges within a constrained time frame. We have reached the following conclusions through mathematical experiments: We observe that the MS– CUR heuristic method gives better results than the DR– CUR heuristic method for the CUR– TCMSP ; see

table (4) for different n . Table (5) shows that although the results of the two proposed approaches are identical, MS- CUR performs better at CPU time for n different durations. Furthermore, both approaches were able to solve all problems when $n \leq 1000$, but failed when $n > 1000$.

Note: In each example, the values of p_j and d_j are produced randomly based on the uniform distribution s.t. $p_j \in [1,10]$ and $d_j \in [1,70]$, according to the condition $p_j \leq d_j$, when $j = 1:n$.

$\sum U_j + R_L$. Also, the submitted work demonstrates that the OPS for problem (CUR-TO) is if and only if one of the EFS for the problem (CUR-TC). Finally, we propose two heuristics for solving search problems, which are fast ways to find (near-optimal) answers to optimization problems.

Table 4: compares the results of using MS-CUR and DR-CUR to find the optimal solution for different values of n .

m	MS- CUR			DR- CUR		
	Av (F_{CUR-TC})	AT/S	ANS	Av (F_{CUR-TC})	AT/S	ANS
4	(55.5, 2.9,14.2)	R	3.0	(58.9, 3.1,14.9)	R	2.0
5	(73.6, 5.0,20.1)	R	4.4	(76.6, 4.7,21.2)	R	2.8
6	(85.4, 5.6,20.7)	R	4.8	(92.3, 6.5,23.1)	R	2.2
7	(144.7, 5.3,32.9)	R	6.0	(154.5, 5.8,32.6)	R	3.0
8	(174.5, 6.2,41.5)	R	6.4	(189.9, 6.6,41.0)	R	3.4
9	(203.8, 7.5,43.8)	R	7.0	(223.9, 9.1,38.8)	R	3.4
10	(256.9, 8.3,46.0)	R	5.6	(280.3, 8.4,44.1)	R	3.0
40	(3071.3, 33.0,183.7)	R	12.4	(3568.4, 34.3,167.5)	R	4.4
70	(10183.6, 44.4,362.9)	R	12.4	(11429.6, 46.5,342.0)	R	3.8
100	(20553.4, 74.4,520.0)	R	14.2	(21929.4, 66.7,494.4)	R	3.6
400	(327276.8, 162.0,2173.3)	1.9	14.0	(365665.4,143.1,2128.5)	2.4	2.8
700	(1021727.1, 413.5,3851.6)	5.0	13.8	(1105138.4,414.4,3788.8)	13.2	2.2
1000	(2040377.5, 225.2,5479.1)	13.4	13.0	(2136874.9,228.6,5401.4)	34.0	2.0

Table 5: Comparison between MS- CUR, DR- CUR for the CUR-TO problem, for different n .

m	MS- CUR			DR- CUR		
	Av (F_{CUR-TO})	AT/S	ANS	Av (F_{CUR-TO})	AT/S	ANS
4	72.6	R	3.0	72.6	R	2.0
5	98.7	R	4.4	98.7	R	2.8
6	111.7	R	4.8	111.7	R	2.2
7	182.9	R	6.0	182.9	R	3.0
8	222.2	R	6.4	222.2	R	3.4
9	255.1	R	7.0	255.1	R	3.4
10	311.4	R	5.6	311.4	R	3.0
40	3288	R	12.4	3288	R	4.4
70	10590.9	R	12.4	10590.9	R	3.8
100	21147.8	R	14.2	21147.8	R	3.6
400	329612.1	1.9	14.0	329612.1	2.4	2.8
700	1025992.2	5.0	13.8	1025992.2	13.2	2.2
1000	2046081.8	13.4	13.0	2046081.8	34.0	2.0

8. CONCLUSIONS

In this work, it has been proven that the specified rules provide efficient (optimal) solutions to the difficulties (CUR-TC and CUR-TO), and they are solved directly without the need for mathematical programming techniques, BAB, or CEM methodology, after a mathematical model is constructed to deal with the research problems $1/(\sum C_j, \sum U_j, R_L)$, $1/\sum C_j +$

ACKNOWLEDGMENTS

The author would like to thank the reviewers of this work for their helpful comments and suggestions that contributed to the improvement of the work.

REFERENCES

- Ezugwu, A. E. S. (2024). Metaheuristic Optimization for Sustainable Unrelated Parallel Machine Scheduling: A Concise Overview With a Proof-of-Concept Study. *IEEE Access*, 12(January), 3386–3416. <https://doi.org/10.1109/ACCESS.2023.3347047>
- 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>
- Khraibet, T. J., & Ghafil, W. K. (2021). Using hybrid GA-PSO algorithm to solve problem in machine scheduling. *Journal of Discrete Mathematical Sciences and Cryptography*, 24(7), 2027–2035. <https://doi.org/10.1080/09720529.2021.1958998>
- Khraibet, T. J., Kalaf, B. A., & Mansoor, W. (2025). A new metaheuristic algorithm for solving multi-objective single-machine scheduling problems. *Journal of Intelligent Systems*, 34(1). <https://doi.org/10.1515/jisys-2024-0373>
- Liao, C. J., & Huang, R. H. (1991). An algorithm for minimizing the range of lateness on a single machine. *Journal of the Operational Research Society*, 42(2), 183–186. <https://doi.org/10.1057/jors.1991.32>
- Naghham M. Neamah and Bayda A. Kalaf. (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>
- Naghham M. Neamah and Bayda A. Kalaf. (2024a). A Hybrid Approach for Efficiently Solving Multi-Criteria Scheduling Problems on a Single Machine. *Iraqi Journal of Science*, 65(12), 7117–7129. <https://doi.org/10.24996/ijs.2024.65.12.26>
- Naghham M. Neamah and Bayda A. Kalaf. (2024b). Solving the Multi-Criteria Problem: Total Completion Time, Total Late Work, Total Earliness Time, Maximum Earliness, and Maximum Tardiness. *Iraqi Journal of Science*, 65(5), 2724–2735. <https://doi.org/10.21533/pen.v11i3.3559.g1288>
- Naghham M. Neamah, & Bayda A. Kalaf. (2024). Solving tri-criteria: total completion time, total late work, and maximum earliness by using exact, and heuristic methods on single machine scheduling problem. *Iraqi Journal for Computer Science and Mathematics*, 5(3), 14–25. <https://doi.org/10.52866/ijcsm.2024.05.03.002>
- Neufeld, J. S., Schulz, S., & Buscher, U. (2023). A systematic review of multi-objective hybrid flow shop scheduling. *European Journal of Operational Research*, 309(1), 1–23. <https://doi.org/10.1016/j.ejor.2022.08.009>
- Tapan S., Farhad M. E., P. D. (1988). A Branch-and-Bound Approach to the Bicriterion Scheduling Problem Involving Total Flowtime and Range of Lateness. *Management Science*, 34(2), 254–260.