Journal of Iraqi Al-Khwarizmi (JIKh) Volume:7 Issue:1 Year: 2026 pages: 230-244

Multi-Criteria Scheduling Problems to Minimize Maximum Late of work Subject to Maximum Earliness and Tardiness with Due Windows

Authors Names

Dhuha Athal Mohsin[a,*]
Hussam Abid Ali Mohammed[b]

Publication date: 30/6/2026

Keywords: *Multi-Criteria, Single-Machine, Due-Windows, Earliness, Tardiness, Late of Work, Lexicographic optimization*

ABSTRACT

This study addresses a single-machine scheduling problem within a multi-criteria optimization framework under due-window constraints, belonging to the class of np-hard problems. each job is associated with a flexible due-window defined by lower and upper bounds, providing a realistic representation of practical production and service environments.

The proposed model considers a hierarchical objective function of the form $1\left|\left[d_i^{(1)}, d_i^{(2)}\right]\right| Lex(V_{max}, ET_{max})$, where the primary objective is to minimize the maximum late work (V_{max}), followed by minimizing the maximum earliness–tardiness (ET_{max}) without affecting the optimality of the first objective. this lexicographic structure reflects priority-based decision-making in real-world scheduling systems. To address the complexity of the problem, several special cases related to the primary objective are analyzed, and a set of dominance properties is developed to reduce the solution space while preserving optimality. these properties are inspired by classical sequencing rules such as minimum slack time (MST) and earliest due date (EDD), adapted to the hierarchical multi-objective context.

The results show that the proposed approach generates high-quality solutions efficiently, achieving performance comparable to complete enumeration methods with significantly lower computational effort. the study also highlights its applicability in just-in-time systems and suggests extensions to more complex scheduling environments.

1. Introduction

In scheduling theory, a due-window is defined as a permissible time interval within which jobs can be completed without incurring penalties. If a job is finished before the beginning of this interval, it is classified as early and a corresponding penalty is applied. Conversely, if completion occurs after the end of the interval, the job is considered tardy and is also penalized. Compared with strict due-date models, this representation better captures real-world situations in manufacturing and service systems, where a certain degree of flexibility in delivery times is usually allowed. The problem addressed in this study falls within the category of scheduling models based on due-date considerations [1].

The Just-in-Time (JIT) production philosophy emphasizes eliminating waste and reducing inventory to minimal levels [2]. Within this framework, completing jobs either ahead of or beyond the specified time leads to unnecessary costs, which are viewed as inefficiencies. Therefore, achieving completion within the assigned time window is considered the most desirable outcome [3].

A considerable body of research has focused on scheduling problems involving due-windows [4]. In some models, the parameters of the window—its location and length—are assumed to be fixed in advance. In other models, these parameters are treated as decision variables and optimized alongside

---

[a]Department of Mathematics, College of Education for Pure Sciences, University of Kerbala, Karbala, Iraq, doaha.a@s.uokerbala.edu.iq

[b]Department of Mathematics, College of Education for Pure Sciences, University of Kerbala, Karbala, Iraq, hussam.abidali@uokerbala.edu.iq

job sequencing; such problems are known as due-window assignment problems. Detailed surveys covering due-date and due-window assignment models are provided in [4], [5], [6]. For example, [4] reviews formulations where earliness and tardiness penalties are represented by general non-decreasing functions, as explored in [7], [8], [9]. The earliest studies introducing due-window concepts can be traced to [10], while more recent contributions addressing common and flexible due-windows appear in [11], [12], [13], [14].

In these models, penalties may be influenced not only by early or late completion but also by the positioning and length of the time window. For instance, [15] presented a polynomial-time solution for a single-machine scheduling problem in which both the starting point and the size of the window are optimized simultaneously. Earlier work in [16] examined cases where either the starting time or the ending time of the window is adjustable, with the goal of minimizing total penalties. On the other hand, [17] analyzed scenarios where the window length is fixed and only its optimal placement is determined.

The problem of assigning a common due-window for all jobs has been studied for both single-machine and parallel-machine environments in [17], [18] respectively, under the assumption of a fixed window size. Subsequent studies, including [15], [19], [20] extended this framework by allowing both the window size and its location to vary. Additional extensions incorporating flow-time penalties were investigated in [21], [22]. Moreover, objective functions based on min–max criteria have been explored in [23], [24] for different machine environments.

More recently, multi-objective optimization approaches have received increasing attention in scheduling research. Among these, lexicographic optimization provides a structured way to handle multiple objectives by assigning priorities rather than combining them into a single function. Under the $Lex(V_{max}, ET_{max})$ criterion, the primary goal is to minimize the maximum variability or deviation measure V_{max}, and once this objective is optimized, the secondary goal is to minimize the maximum earliness–tardiness value ET_{max} without affecting the optimal value of the primary objective. This hierarchical approach is particularly useful in practical settings where certain performance measures are considered more critical than others.

Although many studies in due-window scheduling focus on sum-based or max-based objective functions, relatively few works have addressed lexicographic criteria in this context. Existing results suggest that such criteria can provide more balanced and practically meaningful schedules, especially when dealing with conflicting objectives.

In [25], new theoretical results were developed for multi-criteria scheduling in flow shop systems, aiming to minimize both the makespan and the range of lateness. The study demonstrates that the use of dominance properties can significantly improve solution efficiency and reduce computational effort.

The objective of this paper is to investigate the fundamental characteristics of scheduling problems with predefined due-windows and to develop an efficient solution approach under the lexicographic objective $Lex(V_{max}, ET_{max})$.

In the early stages of scheduling research, most studies focused primarily on optimizing a single performance measure, which led to the development of numerous exact and approximation algorithms tailored to single-criterion problems [26]. However, real-world scheduling applications typically involve several performance aspects that must be considered simultaneously. Despite their practical importance, multi-criteria scheduling problems have received relatively limited attention, mainly due to the high computational complexity associated with combinatorial optimization involving multiple objectives. The difficulty increases further when the criteria are conflicting, meaning that improving one objective may deteriorate another [27].

The simplest form of multi-objective scheduling involves two criteria. In such cases, lexicographic optimization, denoted by $Lex(A, B)$, is commonly used to represent a hierarchical structure of objectives. Under this approach, the primary criterion A is optimized first, and then the secondary criterion B is minimized while maintaining the optimal value of A. Problems of this type are often referred to as secondary criterion problems. Additionally, the three-field notation $\alpha|\beta|\gamma$, introduced by Graham et al., is widely used to formally describe scheduling problems [28].

Several performance measures are frequently used in scheduling, including total completion time, total earliness, total tardiness, maximum lateness, maximum earliness, and maximum tardiness. It has been shown that problems of the form $1||Lex(f, g)$ can be solved in polynomial time when f is the primary objective and g is selected from a specific class of secondary criteria [29]. Such problems can also be viewed as a special case of the more general formulation $1||F(f, g)$, which is typically more complex and may be classified as NP-hard.

In general, three main approaches are used to address multi-criteria scheduling problems. The first approach combines multiple criteria into a single objective using weighted sums. The second treats some criteria as constraints while optimizing others. The third approach generates all Pareto-efficient (non-dominated) solutions and allows decision-makers to select the most appropriate one based on trade-offs [30]. The present study adopts the second approach.

This paper focuses on a single-machine multi-criteria scheduling problem. A set of independent jobs is to be processed on a machine that can handle only one job at a time and is continuously available from time zero. Each job has a known processing time and a specified due-windows, and schedules must ensure that jobs do not overlap. The performance of a schedule is evaluated using multiple cost functions, some of which are non-decreasing with respect to completion times, while others, such as E_{max} may not exhibit regular behavior. These criteria are handled according to a hierarchical structure using the lexicographic formulation $1\left|\left[d_i^{(1)}, d_i^{(2)}\right]\right|Lex(V_{max}, ET_{max})$ where each objective is optimized sequentially while preserving the optimal values of higher-priority objectives.

The remainder of the paper is organized as follows. Section 2 introduces the notation and fundamental concepts. Section 3 presents the mathematical formulation and Special cases. Finally, Section 4 provides the main conclusions and future works.

1.1 Important notations

n: Number of jobs.

p_i: Processing time of job .

$d_i^{(1)}$: The start time of Due-window of job i .

$d_i^{(2)}$: The end time of Due-window of job i .

C_i: Completion time of job i.

s_i: Slack time of job i s.t. $s_i = d_i^{(1)} - p_i$.

$E_i = \max\left\{d_i^{(1)} - C_i, 0\right\}$ the earliness value of the job i.

$E_{max} = \max\{E_i\}$, the maximum value of the earliness.

$T_i = \left\{C_i - d_i^{(2)}, 0\right\}$ the tardiness value of the job i.

$T_{max} = \max\{T_i\}$, the maximum value of the tardiness.
$V_i = \min\{T_i, p_i\}$, the late work value of the job i.
$V_{max} = \max\{\min\{T_i, p_i\}\}$, the maximum value of the late work.

2. Basic Concepts of Combinatorial Optimization Problems

In this study, the following sequencing rules and fundamental concepts are adopted:

Definition 1 [31]**.** The single-machine problem of minimizing the maximum earliness without allowing idle time, denoted by $1||E_{max}$, is optimally solved by applying the Minimum Slack Time (MST) rule. In this rule, jobs are arranged in non-decreasing order of their slack times, where $(s_i = d_i^{(1)} - p_i)$.

Definition 2 [31]**.** The problem of minimizing the maximum tardiness on a single machine with no idle time, represented as $1||T_{max}$, is optimally solved using the Earliest Due Date (EDD) rule. This rule sequences jobs in non-decreasing order of their due-window upper bounds $d_i^{(2)}$, such that $\left(d_1^{(2)} \leq d_2^{(2)} \leq \cdots \leq d_n^{(2)}\right)$.

Definition 3 [32]**.** The Longest Processing Time (LPT) rule arranges all jobs in descending order according to their processing times, i.e., $(p_1 \geq p_2 \geq \cdots \geq p_n)$.

Definition 4 [33]**.** In multi-objective decision-making, the concept of optimization refers to identifying a solution for which no objective can be improved without causing deterioration in at least one other objective.

Theorem 1. If the unconstrained scheduling problem $1||f$ is NP-complete for a given performance measure f, then the corresponding hierarchical problem $1||Lex(f, g)$ remains NP-complete regardless of the choice of the secondary criterion g.

A feasible schedule θ is considered Pareto optimal (or non-dominated) with respect to two performance measures f and g if there exists no other feasible schedule π such that $f(\pi) \leq f(\theta)$ and $g(\pi) \leq g(\theta)$, with at least one of these inequalities being strict [34].

Assume that two performance criteria, f and g are selected for evaluation [35]. When one of these criteria, say f, has higher priority, a natural strategy is to first determine its optimal value, denoted by f^*. Then, among all schedules achieving this optimal value, the one that yields the best performance with respect to g is chosen. This procedure is known as hierarchical or lexicographic optimization. In this framework, the primary criterion f is optimized in the first stage, and the secondary criterion g is subsequently minimized under the constraint that $f = f^*$. The ordering in the Lex expression reflects the priority of the criteria, with the first being the most important [36].

Definition 5 [37]**.** A solution s is said to dominate another solution s' if the corresponding objective vector $z = f(s)$ dominates $z' = f(s')$; that is, $f_i(s) \leq f_i(s')$ for all i, and $f_i(s) < f_i(s')$ for at least one index i.

Theorem 2 [31]**.** The problem $1||f_{max}$ can be solved by iteratively assigning jobs as follows: at each step, select the job that yields the minimum cost when placed in the last available position, and assign it to that position. This process is repeated until all jobs are scheduled.

Lawler Algorithm (LA) [33]**:**

Step (1): Let $N = \{1, 2, \dots, n\}$, initialize the sequence $\omega = \emptyset$, and define M as the set of jobs that have no successors.

Step (2): Select a job $j^* \in M$ such that $f_{j^*}(\sum p_i) = Min\{f_j(\sum p_i)\}, j \in M$. Remove j^* from N, and assign it to the last position in the sequence ω. Update the set M accordingly.

Step (3): If N becomes empty, terminate the procedure; otherwise, return to Step (2). Finally, the following results are obtained:

3. Model Formulation

Let $N = \{1,2,\dots,n\}$ denote a set of jobs to be processed on a single machine. Each job (i) is associated with a processing time (p_i) and a due-window $\left[d_i^{(1)}, d_i^{(2)}\right]$ where $d_i^{(1)} \le d_i^{(2)}$. All jobs are available at time zero, and the machine processes them continuously without interruption. Therefore, a schedule is fully determined by the processing order of the jobs.

For any given sequence σ, let C_i represent the completion time of job (i). The performance measures are defined as follows: the earliness $E_i = max\{0, d_i^{(1)} - C_i\}$ the tardiness $T_i = max\{0, C_i - d_i^{(2)}\}$, and the late work $V_i = min\{T_i, p_i\}$. The due-window size is given by $D_i = d_i^{(2)} - d_i^{(1)}$.

Under the slack due-window approach, the bounds of the due-window are expressed as:

$$d_i^{(1)} = p_i + q^{(1)} \qquad (1)$$

$$d_i^{(2)} = p_i + q^{(2)} \qquad (2)$$

where $q^{(1)}$ and $q^{(2)}$ are constants independent of the jobs and satisfy $q^{(2)} > q^{(1)}$. Consequently, the window size $D_i = q^{(2)} - q^{(1)}$ is identical for all jobs [38].

The objective measures are defined as:

$V_{max} = max\{min\{T_i, p_i\}\}$, which represents the primary objective f, while $E_{max} = max\{E_i\}$, $T_{max} = max\{T_i\}$ and $ET_{max} = E_{max} + T_{max}$ represents the secondary objective g.

Hierarchical Problems:

In this section, we describe the mathematical formulations and solution procedures for problems in which one of the two criteria $Lex(V_{max}, ET_{max})$ is given higher priority than the others. Such problems are referred to as hierarchical or secondary-criteria problems, where the remaining criteria are treated as less important. The formulation of multi-criteria problems follows the same structure as single-criterion models, with additional constraints imposed to preserve the optimal value of the primary objective.

Consider a hierarchical multi-criteria problem of the form: $1||Lex(f, g)$

The formulation can be viewed as consisting of two components:

- The primary objective function f, which is optimized first.
- The secondary objective function g, which is optimized while maintaining the optimal value of f.

Accordingly, solving this type of problem requires two main stages:

Step (1): Determine the optimal value of the primary objective f.

Step (2): Optimize the secondary objective g subject to the condition that the value of f remains unchanged.

For instance, if V_{max} is considered the most significant criterion compared to T_{max} and E_{max}, then the problem $1\left|\left|\left[d_i^{(1)}, d_i^{(2)}\right]\right|\right|Lex(V_{max}, ET_{max})$ can be expressed by minimizing E_{max} while enforcing the constraint $V_{max} = \Delta$, where $\Delta = V_{max}(LA)$.

Additionally, the following condition is imposed: $T_{max} \leq T^*$, $T^* \in \left(T_{max}(LA), T_{max}(MST)\right)$ [33]. This structure ensures that the highest-priority objective is satisfied first, and the remaining criteria are improved without violating it.

$$\left.\begin{array}{l} min\ ET_{max} \\ \ s.t. \\ V_{max} = V_{max}(LA) \end{array}\right\} \qquad (P)$$

Where

$$E_i = \begin{cases} 0, & d_i^{(1)} \leq C_i \quad i = 1,2,\dots,n \\ d_i^{(1)} - C_i, & d_i^{(1)} > C_i \quad i = 1,2,\dots,n \end{cases},$$

$$T_i = \begin{cases} 0, & d_i^{(2)} \geq C_i \quad i = 1,2,\dots,n \\ C_i - d_i^{(2)}, & d_i^{(2)} < C_i \quad i = 1,2,\dots,n \end{cases},$$

$$V_i = \begin{cases} 0, & d_i^{(2)} \geq C_i & i = 1,2,\dots,n \\ C_i - d_i^{(2)}, & d_i^{(2)} < C_i < d_i^{(2)} + p_i & i = 1,2,\dots,n \\ p_i, & d_i^{(2)} + p_i \leq C_i & i = 1,2,\dots,n \end{cases}$$

$E_{max} = max_i\{E_i\}, \qquad i = 1,2,\dots,n$

$T_{max} = max_i\{T_i\}, \qquad i = 1,2,\dots,n$

$V_{max} = max_i\{V_i\} = max_i\{min\{T_i, p_i\}\}, i = 1,2,\dots,n$

E^*_{max} = optimal value obtained using the MST rule.

T^*_{max} = optimal value obtained using the EDD rule.

V^*_{max} = optimal value obtained using the LA rule.

The problem $1\left|\left[d_i^{(1)}, d_i^{(2)}\right]\right| Lex(V_{max}, ET_{max})$ is NP-hard since the problem $1||ET_{max}$ is NP-hard.

4. Special Cases:

Case 1. If $d_i^{(1)} \geq C_i$ for all $i \in N$, then the MST- sequence constitutes an Efficient Solution (ES) for problem P, as illustrated.

Proof. Under the condition $d_i^{(1)} \geq C_i$ for every job $i \in N$ it follows that all jobs are completed before the beginning of their due windows. Hence, all jobs are classified as not late, which implies: ($T_i = 0,$ and $V_i = 0, \forall\, i \in N$. So $T_{max} = 0$ and $V_{max} = 0$). Moreover, the earliness of each job is given by: $E_i = \{d_i^{(1)} - C_i, \forall i \in N\}$. Therefore, $E_{max} = \max\left\{d_i^{(1)} - C_i\right\}, \forall i \in N$.

Consequently, problem P reduces to: $Lex(V_{max}, ET_{max}) = Lex(0, E_{max}) = Lex\left(0, \max\left\{d_i^{(1)} - C_i\right\}\right)$.

Since the MST rule is optimal for minimizing E_{max} ،it directly follows that the MST sequence is an ES for problem P.

Case 2. Assume that $d_i^{(2)} < C_i$ for all $i \in N$, while $\left(C_1 \in \left[d_1^{(1)}, d_1^{(2)}\right]\right)$. Under this setting, three subcases can be distinguished:

a. If $d_i^{(2)} < C_i < d_i^{(2)} + p_i \ \forall i \in N\backslash\{1\}$, then the EDD sequence is an ES.

b. If $d_i^{(2)} + p_i \leq C_i \ \forall i \in N\backslash\{1\}$, then the EDD sequence is also an ES.

c. If the EDD sequence and LA have the same order for all the jobs then this sequence is an ES.

Proof (a). Given that $d_i^{(2)} < C_i$ for all $i \in N\backslash\{1\}$, it follows that no job (except possibly job 1) is early. Thus: $E_i = 0, \forall i \in N$, so $E_{max} = 0$.
For the primary objective when $d_i^{(2)} < C_i < d_i^{(2)} + p_i$ then, $V_i = \min\{T_i, p_i\} = T_i = C_i - d_i^{(2)}, \forall i \in N\backslash\{1\}$, therefore $V_{max} = T_{max}$.
For the secondary objective $ET_{max} = T_{max} = max\left\{max_{i\in N}\left\{C_i - d_i^{(2)}, 0\right\}\right\} = max_{i\in N}\left\{C_i - d_i^{(2)}\right\} = V_{max}$.
Thus the problem becomes: $Lex(V_{max}, ET_{max}) = Lex(V_{max}, V_{max}) = Lex(T_{max}, T_{max}) = Lex\left(max_{i\in N}\left\{C_i - d_i^{(2)}\right\}, max_{i\in N}\left\{C_i - d_i^{(2)}\right\}\right)$.
Since the EDD rule is optimal for minimizing T_{max}, it follows that it is an ES.
Proof (b). Assume that σ is the EDD sequence. If $d_i^{(2)} + p_i \le C_i \ \forall i \in N\backslash\{1\}$, then clearly $d_i^{(2)} < C_i$, which again implies that all jobs are tardy that means $E_i = 0, \ \forall i \in N$. So $E_{max} = 0$.
For the primary objective: $V_i = min\{T_i, p_i\} = p_i, \forall i \in N\backslash\{1\}$, therefore $V_{max} = p_{max}$.
For the secondary objective: $ET_{max} = T_{max} = max\left\{max_{i\in N}\left\{C_i - d_i^{(2)}, 0\right\}\right\} = max_{i\in N}\left\{C_i - d_i^{(2)}\right\}$.
Hence, the objective function reduces to: $Lex(V_{max}, ET_{max}) = Lex(p_{max}, T_{max}) = Lex\left(p_{max}, max_{i\in N}\left\{C_i - d_i^{(2)}\right\}\right)$. Therefore, the EDD sequence is (ES).
Proof (c). Since $d_i^{(2)} < C_i$ and $\left(C_1 \in \left[d_1^{(1)}, d_1^{(2)}\right]\right)$ for all $i \in N\backslash\{1\}$, it follows that: $E_i = 0, \ \forall i \in N$. So $E_{max} = 0$.
Let σ denote the EDD sequence. So $T_{max}(\sigma)$ is optimal value and if $\sigma = LA$ then $V_{max}(\sigma)$ is optimal value in this sequence.
For the primary objective: $V_{max}(\sigma) = \max_{i\in N\backslash\{1\}}\left\{\min\left\{C_i - d_i^{(2)}, p_i\right\}\right\}$.
For the secondary objective, we distinguish: $ET_{max}(\sigma) = T_{max}(\sigma) = \max_{i\in N}\left\{C_i - d_i^{(2)}\right\}$. Hence, the objective function reduces to:
$Lex(V_{max}(\sigma), ET_{max}(\sigma)) = Lex\left(\max_{i\in N\backslash\{1\}}\left\{\min\left\{C_i - d_i^{(2)}, p_i\right\}\right\}, T_{max}\right) =$
$Lex\left(\max_{i\in N\backslash\{1\}}\left\{\min\left\{C_i - d_i^{(2)}, p_i\right\}\right\}, \max_{i\in N\backslash\{1\}}\left\{C_i - d_i^{(2)}\right\}\right)$. Therefore, (σ) sequence is an ES for the problem P.

Case 3. Assume that σ' is any feasible sequence satisfying $\left(d_i^{(1)} \le C_i \le d_i^{(2)}\right)$ for all $i \in N$. In this situation, all jobs are completed exactly within their respective due windows, i.e. they follow a Just-In-Time (JIT) pattern.
Proof. Since each job is processed in their due window, then: $(E_i = 0, \forall\, i \in N$, so $E_{max} = 0)$ and $(T_i = 0\,, \forall\, i \in N$, so $T_{max} = 0)$, which implies: $ET_{max} = 0$
Similarly: $V_i = 0\,, \forall\, i \in N$, so $V_{max} = 0)$. Hence, the objective function reduces to: $Lex(V_{max}, ET_{max}) = Lex(0,0) = (0,0)$.
Therefore, any sequence satisfying the above condition is an ES for problem P.
Proposition 1. If the sequences generated by the MST, EDD and LA rules coincide, then this common sequence yields a unique ES for problem P.
Proof. Let σ denote the sequence such that: $\sigma = MST = EDD = LA$.

The LA rule minimizes V_{max}, making the schedule optimal in terms of late work (Theorem 2). Therefore, the first function has been preserved at its optimal value.

The MST rule minimizes E_{max}, ensuring optimality with respect to earliness (Definition 1). Furthermore

The EDD rule minimizes T_{max}, guaranteeing optimal tardiness performance (Definition 2).

To prove uniqueness, consider any alternative schedule α Then: $V_{max}(\sigma) \leq V_{max}(\alpha)$, $E_{max}(\sigma) \leq E_{max}(\alpha)$ and $T_{max}(\sigma) \leq T_{max}(\alpha)$.

Thus, the pair $(V_{max}(\sigma), ET_{max}(\sigma))$ dominates $(V_{max}(\alpha), ET_{max}(\alpha))$ which establishes that σ is the unique ES for the problem (P).

Case 4. Consider the problem (P) with a common due window $\left[d_i^{(1)}, d_i^{(2)}\right] = \left[d^{(1)}, d^{(2)}\right], \forall i \in N$ and suppose that $d^{(1)} \geq C_i$. If the LPT rule with MST sequence have the same order is applied, then: $Lex(V_{max}, ET_{max}) = Lex\left(0, d^{(1)} - C_1\right)$ and this sequence is an ES.

Proof. Since $d^{(1)} \geq C_i, \forall i \in N$ then all the jobs are completed not late, yielding: ($T_i = 0$ and $V_i = 0, \forall i \in N$. So $T_{max} = 0$ and $V_{max} = 0$) respectively.

So, the first function $V_{max} = 0$ is already hold, it is an optimal value.

With identical due windows, the earliness becomes: $E_i = d^{(1)} - C_i, \forall i \in N$ and therefore: $E_{max} = max\left\{max\{d^{(1)} - C_i, 0\}\right\} = max\{d^{(1)} - C_i\} = d^{(1)} - C_1$

Because MST minimizes E_{max}, it follows that:

$Lex(V_{max}, ET_{max}) = Lex\left(0, d^{(1)} - C_1\right)$ and the sequence is efficient.

Lemma 1. For the objective function $Lex\left(0, d^{(1)} - C_1\right)$ with $d^{(1)} > C_i$ and $LPT = MST, \forall i \in N$, then ES simplifies to: $\left(0, d^{(1)} - p_{max}\right) = \left(0, d^{(1)} - p_1\right)$.

Proof. This follows directly from the definition of LPT- rule ($p_{max} = p_1 = C_1$).

Case 5. Consider problem (P) with a common due window such that $\left[d_i^{(1)}, d_i^{(2)}\right] = \left[d^{(1)}, d^{(2)}\right], \forall i \in N$. Assume that $d^{(2)} \leq C_i$ for all $i \in N\backslash\{1\}$ while $\left(C_1 \in \left[d^{(1)}, d^{(2)}\right]\right)$. Under these conditions, three distinct situations arise:

a. If $d^{(2)} < C_i < d^{(2)} + p_i, \left(C_1 \in \left[d^{(1)}, d^{(2)}\right]\right)$ for all $i \in N\backslash\{1\}$ then: $Lex(V_{max}, ET_{max}) = Lex\left(C_n - d^{(2)}, C_n - d^{(2)}\right)$ and the EDD sequence is an ES.

b. If $d^{(2)} + p_i \leq C_i, \left(C_1 \in \left[d^{(1)}, d^{(2)}\right]\right)$, for all $i \in N\backslash\{1\}$, then: $Lex(V_{max}, ET_{max}) = Lex\left(p_{max}, C_n - d^{(2)}\right)$.

c. If both conditions occur across different jobs in $i, j \in N\backslash\{1\}$ and if the EDD sequence and LA have the same order then: $Lex(V_{max}, ET_{max}) = Lex\left(max_{i \in N\backslash\{1\}}\left\{min\left\{\{C_i - d^{(2)}\}, p_i\right\}\right\}, C_n - d^{(2)}\right)$ and this sequence is an ES.

Proof (a). Since $d^{(2)} < C_i$ for all $i \in N\backslash\{1\}$, it follows that no job (except possibly the first) is completed early. Hence: $E_i = 0$ for all $i \in N$, therefore $E_{max} = 0$.

For the first objective: under the condition $d^{(2)} < C_i < d^{(2)} + p_i$ then: $V_i = \min\{T_i, p_i\} = T_i = C_i - d^{(2)}$, which leads to: $V_{max} = T_{max} = \max\{C_i - d^{(2)}, 0\} = \max\{C_i - d^{(2)}\} = C_n - d^{(2)}$. Thus, $V_{max}(EDD) = C_n - d^{(2)}$ is optimal value.

For the second objective, we have: $ET_{max} = T_{max} = C_n - d^{(2)}$. Also, $T_{max}(EDD) = C_n - d^{(2)}$ is optimal value.
Accordingly, the objective function becomes: $Lex(V_{max}, ET_{max}) = Lex\left(C_n - d^{(2)}, C_n - d^{(2)}\right)$, which is an ES in EDD sequence.
Proof (b). If $d^{(2)} + p_i \le C_i$ for all $i \in N\backslash\{1\}$, then all jobs are tardy, implying:
$E_{max} = 0$ and $T_{max} = C_n - d^{(2)}$.
For the first objective: $V_i = min\{T_i, p_i\} = p_i$ for all $i \in N\backslash\{1\}$,which yields:
$V_{max} = max\{V_i\} = max\{p_i\} = p_{max}$,
For the second objective: $ET_{max} = T_{max} = C_n - d^{(2)}$.
Therefore: $Lex(V_{max}, ET_{max}) = Lex\left(p_{max}, C_n - d^{(2)}\right)$.
Proof (c). When both conditions $d^{(2)} < C_i < d^{(2)} + p_i$ and $d^{(2)} + p_j \le C_j$, for different jobs $i, j \in N\backslash\{1\}$, the analysis combines the previous cases.
The first objective remains: $V_{i,j} = \min\left\{T_i, p_j\right\}$.
and therefore: $V_{max} = \max_{i,j \in N\backslash\{1\}}\left\{V_{i,j}\right\} = \max_{i,j \in N\backslash\{1\}}\left\{min\left\{\left\{C_i - d_i^{(2)}\right\}, p_j\right\}\right\}$, and since EDD sequence and LA have the same order then $V_{max}(LA = EDD)$ is an optimal value.
For the second objective: $E_{max} = 0$ and $T_{max} = C_n - d^{(2)}$ then $ET_{max} = T_{max} = C_n - d^{(2)}$.
Thus, the objective function is: $Lex(V_{max}, ET_{max}) = Lex\left(\max_{i,j \in N\backslash\{1\}}\left\{min\left\{\left\{C_i - d_i^{(2)}\right\}, p_j\right\}\right\}, C_n - d^{(2)}\right)$.
Case 6. Consider problem P under the assumption that all jobs have identical processing times, i.e. $p_i = p, \forall i \in N$. Suppose further that: $d_n^{(1)} \ge C_n = np$ and that the schedule is generated according to the MST rule. Then, this sequence constitutes an ES, and the objective function is given by: $Lex(V_{max}, ET_{max}) = Lex\left(0, d_n^{(1)} - np\right)$.
Proof. Let σ denote the sequence obtained by the MST rule, which orders the jobs according to: $(s_1 \le s_2 \le \cdots \le s_n) = \left(d_1^{(1)} - p \le d_2^{(1)} - p \le \cdots \le d_n^{(1)} - p\right)$, such that $s_i = d_i^{(1)} - p$. Since $d_i^{(1)} \ge C_n = np$ for all $i \in N$, it follows that every job is completed before the start of its due window. Consequently ،all jobs are early and: $T_i = 0$ and $V_i = 0, \forall i \in N$, which implies: $T_{max} = 0$ and $V_{max} = 0$.
The earliness of each job is given by: $E_i = d_i^{(1)} - C_i = d_i^{(1)} - ip$
Hence, the maximum earliness is: $E_{max} = max\left\{d_i^{(1)} - C_i, 0\right\} = max\left\{d_i^{(1)} - ip\right\} = d_n^{(1)} - np$.
Therefore, the objective function can be expressed as:
$Lex(V_{max}, ET_{max}) = Lex(0, E_{max}) = Lex\left(0, d_n^{(1)} - np\right)$.
This confirms that the MST sequence is an ES for problem P.
Case 7. Consider problem P under the assumption that all jobs have identical processing times, i.e. $p_i = p, \forall i \in N$ and that $d_i^{(2)} < C_i = ip$,$\forall i \in N\backslash\{1\}$ with $\left(p \in \left[d_1^{(1)}, d_1^{(2)}\right]\right)$. In this setting, three subcases arise when the EDD = LA sequences have the same order is applied:

a. If $d_i^{(2)} < C_i < d_i^{(2)} + p,$ for all $i \in N\backslash\{1\}$, then: $Lex(V_{max}, ET_{max}) = Lex\left(np - d_{max}^{(2)}, np - d_{max}^{(2)}\right)$.

b. If $d_i^{(2)} + p \leq C_i,$ for all$i \in N\backslash\{1\}$, then: $Lex(V_{max}, ET_{max}) = Lex\left(p, np - d_{max}^{(2)}\right)$.

c. If both conditions occur across different jobs, then: $Lex(V_{max}, ET_{max}) = Lex\left(max_{i\in N\backslash\{1\}}\left\{min\left\{\left\{C_i - d_i^{(2)}\right\}, p\right\}\right\}, np - d_{max}^{(2)}\right)$.

Proof (a). Since all processing times are equal, we have: $C_n = \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} p = np.$
Let σ be the sequence generated by the EDD rule, such that:
$d_1^{(2)} \leq d_2^{(2)} \leq \cdots \leq d_n^{(2)}$. Given that $d_i^{(2)} < C_i = ip$, for all $i \in N\backslash\{1\}$, it follows that no job is early. Hence: $E_i = 0, \forall i \in N\ , so\ E_{max} = 0.$

For the first objective: under the condition $d_i^{(2)} < C_i < d_i^{(2)} + p,$ for all $i \in N\backslash\{1\}$, we obtain $V_i = min\{T_i, p_i\} = T_i\ , \forall i \in N,$ which yields: $V_{max} = max\{V_i\} = max\{T_i\} = max\left\{ip - d_i^{(2)}\right\} = np - d_{max}^{(2)}$. Which it is optimal by used EDD sequence,

For the second objective: $ET_{max} = T_{max} = \max\left\{ip - d_i^{(2)}\right\} = np - d_{max}^{(2)}$.

Thus: $Lex(V_{max}, ET_{max}) = Lex\left(np - d_{max}^{(2)}, np - d_{max}^{(2)}\right)$.

Proof (b). Under the condition $d_i^{(2)} + p \leq C_i,$ for all $i \in N$, then all the jobs remain tardy, and: $E_{max} = 0$ and $T_{max} = np - d_{max}^{(2)}$. For the first objective: $V_i = \min\{T_i, p\} = p$, which implies: $V_{max} = \max\{V_i\} = \max\{p\} = p_{max} = p$.

Hence: $Lex(V_{max}, ET_{max}) = Lex\left(p, np - d_{max}^{(2)}\right)$.

Proof (c). If the conditions $d_i^{(2)} < C_i < d_i^{(2)} + p$ and $d_j^{(2)} + p \leq C_j$ for all $i, j \in N\backslash\{1\}$ hold for different jobs, then: The first objective remains unchanged: $V_i = \min\{T_i, p\}$, which implies: $V_{max} = \max\{V_i\} = \max\{\min\{T_i, p\}\}$ and since the EDD and LA rules have the same order then the first function V_{max} is optimal for all jobs i.

For the second objective: $E_{max} = 0$ and $T_{max} = np - d_{max}^{(2)}$.
which leads to: $ET_{max} = T_{max} = np - d_{max}^{(2)}$.
Accordingly, the objective function becomes:
$Lex(V_{max}, ET_{max}) = Lex\left(max_{i\in N\backslash\{1\}}\left\{min\left\{\left\{C_i - d_i^{(2)}\right\}, p\right\}\right\}, np - d_{max}^{(2)}\right)$.
Thus, in all three subcases, the EDD = LA sequence yields an ES for problem P.

Case 8. Consider problem P under the assumptions that all jobs have identical processing times, i.e. $p_i = p$ and share a common due window: $\left[d_i^{(1)}, d_i^{(2)}\right] = \left[d^{(1)}, d^{(2)}\right]$, for all $i \in N$.

If the condition $d^{(1)} > C_i$, holds for every job $i \in N$ then: $Lex(V_{max}, ET_{max}) = Lex\left(0, d^{(1)} - C_1\right)$ and any feasible sequence σ is an ES.

Proof. Let σ be an arbitrary sequence. Since $d^{(1)} > C_i$, for all $i \in N$ each job is completed before the beginning of the common due window. Therefore ،all jobs are early ،which implies: $T_i = 0$ and $V_i = 0, \forall i \in N$. Hence: $T_{max} = 0$ and $V_{max} = 0$.

The earliness of each job is given by: $E_i = d^{(1)} - C_i$

Since $p_i = p$ we have $C_i = ip$ and thus: $E_i = d^{(1)} - ip$.

Accordingly ،the maximum earliness is: $E_{max} = max\{d^{(1)} - C_i, 0\} = d^{(1)} - \min_{i \in N} C_i$.

Because all processing times are equal ،the smallest completion time corresponds to the first job in the sequence, i.e. $\min_{i \in N} C_i = C_1$.

Hence: $E_{max} = d^{(1)} - C_1$.

Therefore, the objective function reduces to: $Lex(V_{max}, ET_{max}) = Lex(0, d^{(1)} - C_1)$.

This shows that any sequence σ is an ES for problem P.

Lemma 2. For the objective value $Lex(0, d^{(1)} - C_1)$ with, $d^{(1)} > C_n$ the corresponding ES simplifies to: $Lex(0, d^{(1)} - p)$.

Proof. The result follows directly from Lemma (1) and the fact that, under identical processing times, the smallest completion time equals p.

Case 9. Consider problem (P) under the assumptions that all jobs have identical processing times, $p_i = p, \forall i \in N$ and share a common due window, $\left[d_i^{(1)}, d_i^{(2)}\right] = [d^{(1)}, d^{(2)}], \forall i \in N$.

Assume further that $d^{(1)} \geq p$ and that the first completion time satisfies $C_1 = p \in [d^{(1)}, d^{(2)}]$. Under these conditions, three subcases arise:

a. If $d^{(2)} < C_i < d^{(2)} + p$, for all $i \in N\backslash\{1\}$ then: $Lex(V_{max}, ET_{max}) = Lex(np - d^{(2)}, np - d^{(2)})$ and any sequence is an ES.

b. If $d^{(2)} + p \leq C_i$, for all $i \in N\backslash\{1\}$ then: $Lex(V_{max}, ET_{max}) = Lex(p, np - d^{(2)})$ and any sequence is also ES.

c. If both conditions occur across different jobs, then: $Lex(V_{max}, ET_{max}) = Lex(\max_{i \in N\backslash\{1\}}\{\min\{T_i, p\}\}, np - d^{(2)})$ and again any sequence is efficient.

Proof (a). Let σ be an arbitrary sequence. Since $d^{(2)} < C_i$, for all $i \in N$, no job is completed early. Therefore: $E_i = 0$, for all $i \in N$, so $E_{max} = 0$. Given that $p_i = p$, the total completion time is: $C_n = \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} p = np$. The tardiness of each job is: $T_i = C_i - d^{(2)} = ip - d^{(2)}$, for all $i \in N$,which implies: $T_{max} = max\{ip - d^{(2)}\} = np - d^{(2)}$.

Under the condition $d^{(2)} < C_i < d^{(2)} + p$, we have: $V_i = min\{T_i, p_i\} = T_i$ and thus: $V_{max} = T_{max} = max\{ip - d^{(2)}\} = np - d^{(2)}$ and any sequence is efficient and the second function is $ET_{max} = T_{max} = np - d^{(2)}$. Hence: $Lex(V_{max}, ET_{max}) = Lex(np - d^{(2)}, np - d^{(2)})$.

Proof (b). Assume that $d^{(2)} + p \leq C_i$ for all $i \in N$. As before, no job is early, so: $E_{max} = 0$.

The tardiness remains: $T_{max} = np - d^{(2)}$. For the first objective: $V_i = min\{T_i, p\} = p , \forall i \in N$.

which leads to: $V_{max} = p$ and for the second function: $ET_{max} = T_{max} = np - d^{(2)}$.

Therefore: $Lex(V_{max}, ET_{max}) = Lex(p, np - d^{(2)})$.

Proof (c). If both conditions $d^{(2)} < C_i < d^{(2)} + p$ and $d^{(2)} + p \leq C_j$ occur across different jobs, then: The first objective remains: $V_{i,j} = \min\{C_i - d^{(2)}, p\} = \min\{ip - d^{(2)}, p\}$ and consequently, $V_{max} = \max\{\min\{ip - d^{(2)}, p\}\}$.

For the second objective: $E_{max} = 0$ and $T_{max} = np - d^{(2)}$, which implies $ET_{max} = T_{max} = np - d^{(2)}$. Thus: $Lex(V_{max}, ET_{max}) = Lex(\max\{\min\{ip - d^{(2)}, p\}\}, np - d^{(2)})$.

In all three subcases, the objective values depend only on aggregate quantities (such as np and $d^{(2)}$) rather than the specific job order, which explains why any sequence qualifies as an ES for problem P.

Lemma 3. Consider the objective function $Lex\left(\max\left\{\min\left\{ip-d^{(2)},p\right\}\right\},np-d^{(2)}\right)$ under the conditions $d^{(2)} < C_i < d^{(2)} + p$ and $d^{(2)} + p \leq C_j$, for all jobs.

Then, the ES is given by: $Lex\left(\min\left\{np-d^{(2)},p\right\},np-d^{(2)}\right)$.

Proof. Since all jobs have identical processing times, the completion time of the last job is $C_n = \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} p = np$ .Thus, the maximum earliness-tardiness component becomes $ET_{max} = C_n - d^{(2)} = np - d^{(2)}$. Given the stated bounds on completion times, the deviation $C_i - d^{(2)}$is restricted by the processing time p. Hence, the first objective component is determined as $V_{max} = \min\left\{np-d^{(2)},p\right\}$, and the second objective is $ET_{max} = np - d^{(2)}$.

Conclusion

This study investigated a single-machine scheduling problem under due-window constraints within a multi-criteria framework. The main focus was placed on a hierarchical objective of the form $Lex(V_{max}, ET_{max})$, where priority is given to minimizing the maximum late work (V_{max}), followed by minimizing the combined earliness–tardiness measure (ET_{max}) without affecting the optimality of the primary objective. This structure reflects a decision-making setting in which the first objective must be achieved optimally before considering improvements in secondary criteria.

To address the computational complexity of the problem, mathematical formulations were developed along with an analysis of several special cases associated with the primary objective. In addition, a set of special cases was introduced to reduce the search space and improve solution efficiency. These rules extend classical sequencing principles, including MST, EDD, LPT, and the Lawler algorithm, to accommodate the hierarchical nature of the problem. Despite the NP-hardness of the model, the proposed approach demonstrates that efficient solutions can be obtained under specific conditions.

Nevertheless, the scope of this work is limited to a deterministic single-machine environment with fixed due-window parameters. More complex and realistic settings, such as multi-machine systems, uncertain processing times, and dynamic job arrivals, are not considered in this study.

The findings provide valuable insights for Just-in-Time (JIT) scheduling environments, where controlling deviations from desired completion intervals is essential for cost efficiency and operational stability. The proposed formulations and dominance properties can support decision-makers in constructing improved schedules with reduced delays and deviations. However, the study remains primarily theoretical, and the practical applicability of the proposed rules in real-world systems has not been empirically evaluated. This represents an important direction for further investigation.

Future Work

Future research may extend this work by exploring additional problem settings through modifications of the objective functions or by incorporating new constraints. For example, extensions may include problems such as $1\left|\left[d_i^{(1)},d_i^{(2)}\right]\right|Lex(E_{max},\sum T_i)$ and $1\left|\left[d_i^{(1)},d_i^{(2)}\right]\right|Lex(\sum C_i, ET_{max})$, which consider alternative performance measures or the inclusion of release times.

Further studies could also investigate more generalized environments, including stochastic and dynamic scheduling models, as well as the development of heuristic and metaheuristic approaches capable of handling large-scale instances efficiently.

References

[1] M. Engineering, P. Engineering, and I. E. Submitted, “Development of heuristics for single processor scheduling problems with tardiness - related performance measures by saheed akande of the university of ibadan,” no. MATRIC. NO.: 153264, pp. 19–30, http://hdl.handle.net/123456789/310, 2017.

[2] J. Józefowska, *Just-In-Time Scheduling: Models and Algorithms for Computer and Manufacturing Systems*, vol. 106. in International Series In Operations Research amp; Mana, vol. 106. Boston, MA: Springer US, 2007. doi: 10.1007/978-0-387-71718-0.

[3] S. Akande, G. O. Ajisegiri, and T. M. Azeez, “Minimizing Conditional Mean Tardiness and Mean Earliness on a Single Machine with the due Windows Approach,” *Eng. Innov.*, vol. 11, no. 2813–1002, pp. 83–98, https://doi.org/10.4028/p-i6uNLi, 2024, doi: 10.4028/p-i6unli.

[4] V. Gordon, J. M. Proth, and C. Chu, “A survey of the state-of-the-art of common due date assignment and scheduling research,” *Eur. J. Oper. Res.*, vol. 139, no. 1, pp. 1–25, https://doi.org/10.1016/S0377-2217(01)00181-, 2002, doi: 10.1016/S0377-2217(01)00181-3.

[5] J.-B. Wang, Y. Hu, and B. Zhang, “Common due-window assignment for single-machine scheduling with generalized earliness/tardiness penalties and a rate-modifying activity,” *Eng. Optim.*, vol. 53, no. 3, pp. 496–512, https://doi.org/10.1080/0305215X.2020.174, Mar. 2021, doi: 10.1080/0305215X.2020.1740921.

[6] T. C. E. Cheng and M. C. Gupta, “Survey of scheduling research involving due date determination decisions,” *Eur. J. Oper. Res.*, vol. 38, no. 2, pp. 156–166, https://doi.org/10.1016/0377-2217(89)9010, Jan. 1989, doi: 10.1016/0377-2217(89)90100-8.

[7] V. S. Gordon, J.-M. Proth, and C. Chu, “Due date assignment and scheduling: Slk, TWK and other due date assignment models,” *Prod. Plan. Control*, vol. 13, no. 2, pp. 117–132, https://doi.org/10.1080/09537280110069621, Jan. 2002, doi: 10.1080/09537280110069621.

[8] A. Federgruen and G. Mosheiov, “Simultaneous optimization of efficiency and performance balance measures in single-machine scheduling problems,” *Nav. Res. Logist.*, vol. 40, no. 7, pp. 951–970, https://doi.org/10.1002/1520-6750(199312), 1993, doi: 10.1002/1520-6750(199312)40.

[9] F. D. Anger, C. Y. Lee, and L. A. Martin-Vega, “Single machine scheduling with tight windows,” *Res. Pap.*, pp. 16–86, 1986.

[10] T. C. E. Cheng, “Optimal common due-date with limited completion time deviation,” *Comput. Oper. Res.*, vol. 15, no. 2, pp. 91–96, https://doi.org/10.1016/0305-0548(88)90001-, Jan. 1988, doi: 10.1016/0305-0548(88)90001-9.

[11] S. D. Liman, S. S. Panwalkar, and S. Thongmee, “Common due window size and location determination in a single machine scheduling problem,” *J. Oper. Res. Soc.*, vol. 49, no. 9, pp. 1007–1010, https://doi.org/10.1057/palgrave.jors.2, Sep. 1998, doi: 10.1057/palgrave.jors.2600601.

[12] A. Janiak, T. Kwiatkowski, and M. Lichtenstein, “Scheduling problems with a common due window assignment: A survey,” *Int. J. Appl. Math. Comput. Sci.*, vol. 23, no. 1, pp. 231–241, DOI: 10.2478/amcs-2013–0018, Mar. 2013, doi: 10.2478/amcs-2013-0018.

[13] S. D. Liman, S. S. Panwalkar, and S. Thongmee, “Determination of common due window location in a single machine scheduling problem,” *Eur. J. Oper. Res.*, vol. 93, no. 1, pp. 68–74, DOI: 10.1016/0377-2217(95)00181, Aug. 1996, doi: 10.1016/0377-2217(95)00181-6.

[14] B. Bai, C.-M. Wei, H.-Y. He, and J.-B. Wang, “Study on Single-Machine Common/Slack Due-Window Assignment Scheduling with Delivery Times, Variable Processing Times and Outsourcing,” *Mathematics*, vol. 12, no. 18, pp. 2883, DOI 10.3390/math12182883, Sep. 2024, doi: 10.3390/math12182883.

[15] A. Janiak and M. Marek, “Single machine scheduling problem with optimal due interval assignment,” *Lect. Notes Silesia Univ. Technol. Autom.*, vol. 129, pp. 145–155, 2000, [Online]. Available:

https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Single+machine+scheduling+problem+with+optimal+due+interval+assignment&btnG=

[16] A. Janiak and M. Marek, "Multi-machine scheduling problem with optimal due interval assignment subject to generalized sum type criterion," in *Operations Research Proceedings 2001: Selected Papers of the International Conference on Operations Research (OR 2001) Duisburg, September 3–5, 2001*, Springer, Berlin, Heidelberg, 2002, pp. 207–212, DOI https://doi.org/10.1007/978-3-642–502. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-50282-8_26

[17] G. Mosheiov and A. Sarig, "A multi-criteria scheduling with due-window assignment problem," *Math. Comput. Model.*, vol. 48, no. 5–6, pp. 898–907, https://doi.org/10.1016/j.mcm.2007.08.018, Sep. 2008, doi: 10.1016/j.mcm.2007.08.018.

[18] W. K. Yeung, C. Oguz, and T. C. E. Cheng, "Single-machine scheduling with a common due window," *Comput. Oper. Res.*, vol. 28, no. 2, pp. 157–175, https://doi.org/10.1016/S0305-0548(99)000, Feb. 2001, doi: 10.1016/S0305-0548(99)00097-0.

[19] G. Mosheiov, "A Due-Window Determination In Minmax Scheduling Problems," *INFOR Inf. Syst. Oper. Res.*, vol. 39, no. 1, pp. 107–123, https://doi.org/10.1080/03155986.2001.117, Feb. 2001, doi: 10.1080/03155986.2001.11732429.

[20] A. Janiak and M. Marek, "Scheduling problems with optimal due interval assignment subject to some generalized criteria," in *Operations Research Proceedings 2002: Selected Papers of the International Conference on Operations Research (SOR 2002), Klagenfurt, September 2–5, 2002*, Springer, 2003, pp. 217–222, DOI https://doi.org/10.1007/978-3-642–555. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-55537-4_35

[21] D. Shabtay, G. Mosheiov, and D. Oron, "Single machine scheduling with common assignable due date/due window to minimize total weighted early and late work," *Eur. J. Oper. Res.*, vol. 303, no. 1, pp. 66–77, https://doi.org/10.1016/j.ejor.2022.02.017, Nov. 2022, doi: 10.1016/j.ejor.2022.02.017.

[22] Rajwan Hamood Sheah and Iraq T. Abbas, "Using Multi-Objective Bat Algorithm for Solving Multi-Objective Non-linear Programming Problem," *Iraqi J. Sci.*, vol. Vol 62, no. No. 3, pp. 997–1015, DOI: https://doi.org/10.24996/ijs.2021.6, Mar. 2021, doi: 10.24996/ijs.2021.62.3.29.

[23] F. H. Ali, R. J. Mitlif, and W. A. Hussein, "Exact and Near Pareto Optimal Solutions for Total Completion Time and Total Late Work Problem," *Iraqi J. Sci.*, pp. 4385–4894, DOI: https://orcid.org/0000-0001-5033–8, Jul. 2023, doi: 10.24996/ijs.2023.64.7.29.

[24] M. H. Ibrahim, F. H. Ali, and H. A. Chachan, "Dominance rules for single machine scheduling to minimize weighted multi objective functions," vol. 2414, no. 1, pp. 040090, DOI: https://doi.org/10.1063/5.0119047, 2023, doi: 10.1063/5.0119047.

[25] S. F. Yousif and F. H. Ali, "Solving Maximum Early Jobs Time and Range of Lateness Jobs Times Problem Using Exact and Heuristic Methods," *Iraqi J. Sci.*, vol. 65, no. 2, pp. 923–937, https://orcid.org/0009-0009-7319–2633, 2024, doi: 10.24996/ijs.2024.65.2.28.

[26] E. Steure Ralph, "M ultiple Criteria Op timization: Theory, Comutation and Plication," *Willey*, 1986.

[27] 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.

[28] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," in *Annals of discrete mathematics*, vol. 5, Elsevier, 1979, pp. 287–326.

[29] C.-Y. Lee and G. L. Vairaktarakis, "Complexity of single machine hierarchical scheduling: A survey," in *Complexity in numerical optimization*, World Scientific, 1993, pp. 269–298.

[30] C. Tsiushuang, Q. Xiangtong, and T. Fengsheng, "Single machine scheduling to minimize weighted earliness subject to maximum tardiness," *Comput. Oper. Res.*, vol. 24, no. 2, pp. 147–

152, 1997.

[31] J. A. Hoogeveen, “Single-machine bicriteria scheduling,” Eindhoven University of Technology (TU/e), Centrum voor Wiskunde en Informatica, 1992. doi: 10.6100/IR367531.

[32] F. Della Croce and R. Scatamacchia, “The Longest Processing Time rule for identical parallel machines revisited,” *J. Sched.*, vol. 23, no. 2, pp. 163–176, DOI https://doi.org/10.1007/s10951-018–05, 2020, doi: 10.1007/s10951-018-0597-6.

[33] K. Alshaikhli and A. Alshaikhli, “Algorithms for Multicriteria Scheduling Problems to Minimize Maximum Late Work, Tardy, and Early,” *J. Appl. Math. Phys.*, vol. 12, no. 02, pp. 661–682, 2024, doi: 10.4236/jamp.2024.122043.

[34] J. Y. T. Leung, *Handbook of scheduling: algorithms, models, and performance analysis*. Chapman and Hall/CRC, 1990.

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

[36] G. W. Evans, “An overview of techniques for solving multiobjective mathematical programs,” *Manage. Sci.*, vol. 30, no. 11, pp. 1268–1282, 1984.

[37] J. E. C. Arroyo, R. Dos Santos Ottoni, and A. De Paiva Oliveira, “Multi-objective variable neighborhood search algorithms for a single machine scheduling problem with distinct due windows,” *Electron. Notes Theor. Comput. Sci.*, vol. 281, no. May 2014, pp. 5–19, DOI: https://doi.org/10.1016/j.entcs.2011.11, 2011, doi: 10.1016/j.entcs.2011.11.022.

[38] B. Cheng and L. Cheng, “Single Machine Slack Due-Window Assignment and Scheduling of Linear Time-Dependent Deteriorating Jobs and a Deteriorating Maintenance Activity,” *OALib*, vol. 05, no. 10, pp. 1–9, DOI: https://doi.org/10.48550/arXiv.1405.0712, 2018, doi: 10.4236/oalib.1104907.