JOBS مجلة العلوم الأساسية
Journal of Basic Science
Print -ISSN 2306-5249
Online-ISSN 2791-3279
العدد الثامن والاربعون
٢٠٢٦م /١٤٤٨هـ
العدد الثامن والاربعون
(645) (625)

اختزال المعايير الرباعية مع وقت الاصدار لمشاكل جدولة الآلة الواحدة

م.د.نغم موسى نعمه

م.د.تهاني جبار خريبط

م.د.شيماء عباس جاسم

جامعة بغداد / كلية العلوم للبنات / قسم الرياضيات[1,3]

وزارة التربية / مديرية تربية ذي قار[2]

naghamm_math@csw.uobaghdad.edu.iq

المستخلص:

في هذا البحث، قمنا بدراسة ومناقشة مسألة جدولة ماكنة واحدة. وقد اقترحنا المسألة الأولى $(\widetilde{G1})$ لجدولة مجموعة من المهام المراد معالجتها على ماكنة واحدة بهدف تقليل كل من: الحد الأقصى للعمل المتأخر V_{max}، والحد الأقصى لوقت التأخير T_{max}، والحد الأقصى لوقت التدفق F_{max}، وإجمالي وقت التدفق $\sum F_g$ في آن واحد؛ ويرمز لها بـ $1/r_g\,/(V_{max}, T_{max}, F_{max}, \sum F_g)$، مع الملاحظة ان لكل مهمة وقت إصدار مختلف r_g. انطلاقًا من المسألة الأولى $(\widetilde{G1})$ ، استنتجنا المسألة الثانية $(\widetilde{G2})$، والتي يُرمز لها $1/r_g\,/\,V_{max} + T_{max} + F_{max} + \sum F_g$. في البداية، وجدنا الصيغة الرياضية للمسألتين، ثم ناقشنا الحالات الخاصة لحلهما مباشرةً دون اللجوء إلى خوارزميات CEM أو BAB أو البرمجة الرياضية. بالإضافة إلى ذلك، فقد ثبت أن الحل الأمثل للمشكلة $(\widetilde{G2})$ هو أحد الحلول الفعالة للمشكلة $(\widetilde{G1})$.

JOBS
مجلة العلوم الأساسية
Journal of Basic Science
Print -ISSN 2306-5249
Online-ISSN 2791-3279
العدد الثامن والاربعون
٢٠٢٦م /١٤٤٨هـ

Reducing the Quadruple-Criteria with Release Time for Single-Machine Scheduling Problems

Lect. Dr.Nagham M. Neamah[1,a)]

Lect. Dr.Tahani Jabbar Khraibet[2]

Lect. Dr.Shaima Abbas Jasim[3]

[1,3]Mathematics Department, College of Science for Women, University of Baghdad, Baghdad, Iraq

[2]Thi-Qar Directorates of Education, Ministry of Education, Iraq

[a)] Corresponding Author Email: naghamm_math@csw.uobaghdad.edu.iq

Abstract: In this research, we have researched and discussed the issue of Single-machine scheduling problems**.** We have scheduling a single machine. proposed the first problem $1/r_g\ /(V_{max}, T_{max}, F_{max}, \sum F_g)$ of scheduling a set of tasks to be processed on a single machine to minimize the maximum (T_{max}), maximum flow time (V_{max}), maximum tardiness time lateness $(\widetilde{G1})$**,** noting $(\sum F_g)$ simultaneously, denoted as (F_{max}), and total flow time that each task has a different release time(r_g). From the first problem $(\widetilde{G1})$, we deduced the second problem$1/r_g\ /V_{max} + T_{max} + F_{max} + \sum F_g$, which is denoted as $(\widetilde{G2})$. First, we found the mathematical formula for the two problems, and then we discussed the special cases for solving the two problems directly without resorting to CEM, BAB, or mathematical programming. In addition, it has been proven that the optimal solution to problem $(\widetilde{G2})$ is one of the effective solutions to problem $(\widetilde{G1})$.

Keywords: Flow Time, Late Work. Single-Machine Scheduling, Release Time, Tardiness Time

1. INTRODUCTION

The machine scheduling problem (MSP) is an issue in operations research that involves optimisation(de Weerdt et al., 2021). The scheduling problem is described as the challenge of allocating a collection of tasks or jobs to a set of resources or machines at a specified time. Scheduling is a key decision-making tool in many fields, including manufacturing, engineering, and commerce, where the objective is to minimise costs and energy consumption while increasing profitability, performance, and efficiency (Baker, 2009). The scheduling problem is crucial in most industrial and manufacturing systems, as well as in many information-processing situations (Nagham M. Neamah and Bayda A. Kalaf, 2024b). It is also utilised in various manufacturing and service industries to decrease (raise) the cost (benefit) of production in the industrial process, allowing businesses to compete, as well as in transmission and distribution settings. Furthermore, scheduling is resource allocation across time to fulfil a set of tasks, and it is a subset of the larger domain of a combinatorial optimisation problem (COP). It is either a minimisation or a maximisation issue, with several related instances (Phrueksanant, 2013). The scheduling issue is sometimes characterised as the arranging of units (people, tasks, cars, lectures, etc.) in space and time in a way that limitations and specific goals are met. In recent decades, the Single-Machine Scheduling Problem (SMSP) has garnered a lot of attention (Johnson, 1954), and it contributed to the initial theoretical development of scheduling. The results were then presented. Until the end of the 1980s, mainstream research focused on a single objective issue. However, in the actual world, practically all choice issues contain specific criteria, which might be many or conflicting (Smith, 1956),(Nagham M. Neamah and Bayda A. Kalaf, 2024d). As a result, scheduling issues are more complex in models with many goals (criteria). It is frequently improbable that the same set of choice factors may best serve several purposes. As a result, in multi-criteria and multi-objective cases, a group of Pareto-optimal solutions (Efficient Solutions) is identified using multi-objective optimisation based on competing objective functions instead of a single optimal solution. MSPs are often processed using optimisation algorithms that consistently find (near) optimum solutions (Allaoua & Brahim, 2017). However, optimisation algorithms cannot be created in polynomial time for all optimisation issues.

Here are a few of the literature reviews: Many academics have examined and confirmed various dominant properties and presented numerous methods for the issue $1/r_j/\sum F_j$, demonstrating that it is NP-hard (Agin, 1966), (Michael L. Pinedo, 1994),(Batsyn et al., 2014),(Daowd et al., 2022). Sankaran performed the first study of a scheduling problem on a single machine, where penalties are imposed on tasks that start before their target start date or are completed after their due date. A more efficient algorithm than the one in question is proposed to reduce the maximum penalty, taking into account certain assumptions about target start times, due dates, and penalty functions. The number of calculations in the proposed algorithm is approximately $nlogn$ (Lakshminarayan et al., 1978). The problem $1/r_j/\sum F_j$, which may be efficiently addressed using the well-known Shortest Processing Time (SPT) rule (Smith, 1956), if all jobs are issued simultaneously. (Ahmadi, 1990) it was proven that the minimum (using SRPT rules) is the optimal solution for scheduling N non-preemptive tasks with unequal release times on a single machine to minimize the sum of task completion times, whereas Chandra et al. (Opérationnelle & Opérationnelle, 1996) created branch-and-bound methods. The problem $\sum F_j + \sum U_j$ has several applications, including network systems and parallel computing (Woeginger, 1999).The proactive form of the issue, $1/r_j, \text{pmtn}/\sum F_j$, can be solved optimally in polynomial time using the shortest residual processing time (SRPT) algorithm. (Chu et al., 2000) presented approximation strategies for solving this problem. (Al-Zuwaini M.K., 2011) demonstrated dominance rules for the $1/r_j/\sum F_j$ issue by effective branching and restriction techniques with an acceptable minimum. Researchers have combined this criterion ($\sum F_j$) with additional criteria, such as $\sum F_j + \sum U_j$ (Al-Zuwaini M.K., 2011) . $1/r_j/ V_{max}$ is an NP-

JOBS مجلة العلوم الأساسية
Journal of Basic Science
Print -ISSN 2306-5249
Online-ISSN 2791-3279
العدد الثامن والاربعون
٢٠٢٦م /١٤٤٨هـ

hard problem. Lawler created the algorithm (LA) and demonstrated that it solves $1// f_{max}$ in $O(n^2)$ time (E . L . Lawler, 2013),(Daowd et al., 2022). Mahdi and Qasim (Taylor et al., 2009) provided a research on the issue $1/r_j /E_{max} + T_{max}$. They proposed the (B&B) algorithm, which applies large-scale effective dominance rules. The approach obtains lower bound (LB) by relaxing the non-interruption assumption and separating the issue into two subproblems: $(1/r_j, pmtn/T_{max})$ and $(1/r_j, pmtn/E_{max})$. The two issues are then addressed by using methods developed from the two rules (EDD and MST). The computer studies indicated that the suggested technique is efficient in addressing problems with up to 1000 tasks.(Cheachan et al., 2019), Using artificial intelligence algorithms to find the optimal solution to the following problem $1/r_k/ \sum_{k=1}^{n} C_k + \sum_{k=1}^{n} U_k + T_{max}$. (Chen et al., 2020) Studying the problems of hierarchical scheduling for a single-machine with release and advance times, where the primary criterion is the total completion time and the secondary criterion is an arbitrarily regular scheduling criterion, which is either the sum form or the maximum form. They are developing new solution technologies that consolidate certain genetic characteristics of possible timescales. (Chachan & Jaafar, 2020), they introduced the Branching and Constraining (B&B) algorithm to reduce the sum of total completion time, total delays, total advances, number of delayed jobs, and total delayed work with unequal release times, six inferential methods were proposed for determining the upper limit of the calculation. Also, to obtain a lower bound (LB) for this problem, they modified a lower bound selected from the literature (Moore's algorithm and Lawler's algorithm), and some rules of dominance were proposed. Two special cases were also deduced.(Grigoreva & Didenko, 2025) presents a new approximate algorithm for the single-machine scheduling problem with release and delivery times, which includes additional conditions requiring a certain delay between the completion of one task and the start of the next (late priority constraints).

In this study, single-machine scheduling with release times is considered. Our goal is to minimize maximum late work, the maximum tardiness time, the maximum flow time, and total flow time of jobs, respectively, and the summation of the total flow time of jobs, the maximum flow time, and maximum late work, respectively, and solve these problems directly without the need to use the exact counting method, branch and bound method, and artificial intelligence algorithms through proof of special cases. In a special case, if it exists, it depends on meeting certain conditions to make the problem solvable easily. The special case, if any, depends on meeting certain conditions to make the problem easy to solve; those conditions are dependent on the target function as well as the functions.

The remainder of this article is organized as follows: In Section 2, we propose mathematical models aimed at addressing the four conflicting criteria in the single- machine scheduling problem. Sections 3 and 4 present specific cases that illustrate how to identify effective (optimal) solutions to a given problem. Section5 presents and discusses the main findings from the preceding sections. The conclusions are presented in Section 6.

2. Mathematical Framework

This section presents the mathematical formulation of the single-machine scheduling problem for four criteria and four objective functions. First, we present some of the symbols used in formulating the four criteria and four objective functions for the single-machine scheduling problem in Table 1:

Table 1: List of symbols and abbreviations

Symbol	Description
BAB	Branch and Bound
CEM	Complete enumeration method
d_g	The due date of job g is the date when the job should ideally be completed.
(EF.S)	Efficient Solution
Z_I	Objective function value for the $(1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)$ problem
Z_{II}	Objective function value for the $(1/r_g/V_{max} + T_{max} + F_{max} + \sum F_g)$ problem
I_g	The idle time, where $I_g = r_g$ if $g = 1$, $I_g = r_g - C_{g-1}$ if $r_g > C_{g-1}, g = 2{:}\, m$, $I_g = 0$ otherwise

max	Maximum
min	Minimize
MSP	Machine scheduling problem
m	Number of Jobs
(OP.S)	Optimal Solution
p_g	Processing time of job g, which means that it has to be processed for a period of length p_g.
r_g	A release time r_g of job g is the earliest time job that g can start to process (the earliest possible start date). If this code does not exist, the job g may start processing at any time. If r_g exists, which means the jobs in MSP may have different release times. Otherwise, all jobs arrive at time zero.
SPT	Shortest processing time, which is sequencing jobs in a non-decreasing manner according to their processing times p_g. This rule can solve the $1//\sum_{g=1}^{m} C_g$ problem (Smith1956).
SRD	Shortest release time, that is, sequencing jobs in a non-decreasing manner of their release time. r_g, which solves $1/r_g/C_{max}$ problem (Michael L. Pinedo, 1994)
SMSP	Single-machine scheduler problem

The work presented deals with scheduling an $g = 1{:}m$ set of n independent tasks on a single machine with a quadruple-criteria function and a quadruple-objective function denoted by $1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g), 1/r_g/V_{max} + T_{max} + F_{max} + \sum F_g$. In this problem, interruption is not allowed, no priority relation is assumed between tasks, and only one task g can be processed at any given time within a sequence of tasks $ð = (ð_1, ð_2, \dots, ð_m)$. Every one of the jobs g has a release time r_g where it can't be processed prior, requires the processing of p_g time units on a machine, and optimally must be completed at its due date d_g. Then, for every task g, we calculate the time of completion by $C_1 = r_{ð1} + p_{ð1}$, $C_{ðg} = max\{r_{ðg}, C_{ð(g-1)}\} + p_{ðg}$ for $g = 2{:}m$. The flow time of task g is defined by $F_{ðg} = C_{ðg} - r_{ðg}$.The tardiness of task g is defined by $T_{ðg} = max\{C_{ðg} - d_{ðg}, 0\}$, and the Late work of task g is defined by $V_{ðg} = min\{T_{ðg}, p_{ðg}\}$.

The objective is to establish a schedule for the minimisation of the Max.Late work V_{max}, Max. Tardiness T_{max}, Max. flow time F_{max}, and total flow time $\sum F_g$. Let ð be the set of all possible solutions, and ð a timeline in $\check{S}$. The mathematical formula for our problems can be written as follows:

The first problem $\left(1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)\right)$ was described as follows:

$$\text{Objective Functions: } Z_I = Min\begin{cases} V_{max} \\ T_{max} \\ F_{max} \\ \sum F_g \end{cases} \quad (1)$$

Subject to

$$\text{Restrictions: }\begin{cases} C_1 = r_{ð_1} + p_{ð_1} & (2) \\ C_{ð_k} \geq r_{ð_k} + p_{ð_k} & (3) \\ C_{ð_k} \geq C_{ð(k-1)} + p_{ð_k} & (4) \\ F_{ð_k} = C_{ð_k} - r_{ð_k} & (5) \\ T_{ð_k} \geq C_{ð_k} - d_{ð_k} & (6) \\ V_{ð_k} \leq p_{ð_k} & (7) \\ V_{ð_k} \leq T_{ð_k} & (8) \end{cases}$$

Non-negative condition: $p_{ð_k}, d_{ð_k} > 0, F_{ð_g} \geq p_{ð_g}, r_{ð_k}, V_{ð_k}, E_{ð_k}, T_{ð_k} \geq 0$ (9)

The second problem $\left(1/r_g/V_{max} + T_{max} + F_{max} + \sum F_g\right)$ was described as follows:

Objective Functions: $Z_{II} = Min\left(V_{max} + T_{max} + F_{max} + \sum F_g\right)$ (10)

Subject to

$$\text{Restrictions: } \begin{cases} C_1 = r_{ð_1} + p_{ð_1} & (11) \\ C_{ð_k} \geq r_{ð_k} + p_{ð_k} & (12) \\ C_{ð_k} \geq C_{ð(k-1)} + p_{ð_k} & (13) \\ F_{ð_k} = C_{ð_k} - r_{ð_k} & (14) \\ T_{ð_k} \geq C_{ð_k} - d_{ð_k} & (15) \\ V_{ð_k} \leq p_{ð_k} & (16) \\ V_{ð_k} \leq T_{ð_k} & (17) \end{cases}$$

Non-negative condition: $p_{ð_k}, d_{ð_k} > 0, F_{ð_g} \geq p_{ð_g}, r_{ð_k}, V_{ð_k}, E_{ð_k}, T_{ð_k} \geq 0$ (18)

The problems $1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)$, and $1/r_g/V_{max} + T_{max} + F_{max} + \sum F_g$ are known as $\breve{G1}$, and $\breve{G2}$ respectively, where $\breve{G1}$ is a quadruple-criteria function, and $\breve{G2}$ is a quadruple-objective function. Finding the efficient (optimal) collection of solutions to these NP-hard issues is challenging. In this paper, we have illustrated particular situations and proved that they offer efficient (optimal) solutions to problems without the need for CEM, BAB, or mathematical programming approaches. This indicates that solving the NP-hard machine scheduling issue (Lee and Liman, 1992), (Woeginger, 1999), ; machine scheduling issue is challenging, and it gets even more challenging when the goal function includes several targets. There are instances when this issue can be solved. The specific case of the scheduling issue is to find an ideal timetable directly without requiring mathematical programming techniques. If there is a special case, it depends on certain requirements being met in order for the problem to be readily solved. The particular case, if any, depends on satisfying specific requirements to make the problem straightforward to solve. Both the target function and the functions determine these conditions.

3. SPECIAL CASES FOR PROBLEMS $\breve{G1}$

Case I: If $r_g = r, p_g = p$ & $d_g = d, \forall g = 1{:}m$, then any schedule provides an (EF.S) for the $\breve{G1}$ problem.

Proof: In any sequence $C_{ð_g} = gp + r,\ F_{ð_g} = -r + C_{ð_g} = gp$, therefore

$\sum_{g=1}^{m} F_{\eth_g} = -\sum_{g=1}^{m} r_{\eth_g} + \sum_{g=1}^{m} C_{\eth_g} = \sum_{g=1}^{m} gp = \left(\frac{m^2+m}{2}\right)p,$
(19)

$F_{max} = mp,$
(20)

Also, $T_{\eth_g} = max\{-(d-(r+gp)),0\}$, and $V_{\eth_g} = min\{T_{\eth_g}, p\} = min\{max\{(r+gp)-d,0\},p\}$, then

$T_{max} = max\{max\{-(d-(r+gp)),0\}\} = max\{(r+mp)-d,0\},$
(21)

$V_{max} = max\{min\{max\{(r+gp)-d,0\},p\}\},$
(22)

there are three cases:

- if $C_{\eth_g} < d, \forall g$ this means that every task must be completed early, then $T_{\eth_g} = V_{\eth_g} = T_{max} = V_{max} = 0$, and from equations (19), (20), hence $(V_{max}, T_{max}, F_{max}, \sum F_m)$ reduce to $(F_{max}, \sum F_g) = \left(mp, \left(\frac{m^2+m}{2}\right)p\right)$.
- if $\le C_{\eth_g} \le d+p, \forall g$, in other words, all the jobs are completed late, hence $T_{max} = r+mp-d = V_{max}$
(23)

And from equations (19), (20), and (23) hence $(V_{max}, T_{max}, F_{max}, \sum F_g) = (r+mp-d, r+mp-d, mp, \left(\frac{m^2+m}{2}\right)p)$.

- if $C_{\eth_g} > d+p, \forall g$, In other words, all jobs are done late, hence

$T_{max} = V_{max} = p,$
(24)

In addition to equations 4 and 5, we obtain $(V_{max}, T_{max}, F_{max}, \sum F_g) = (p, p, mp, \left(\frac{m^2+m}{2}\right)p)$. Since the four quantities are constant. So, any of the schedules is an (EF.S) for the $\widetilde{G1}$ problem.

Case II: If $p_g = p, r_g = r, \forall g$, then any schedule ∂ fulfils $d_g = gp + r$ gives an (EF.S) for the $\widetilde{G1}$ problem.

JOBS مجلة العلوم الأساسية
Journal of Basic Science
Print -ISSN 2306-5249
Online-ISSN 2791-3279
العدد الثامن والاربعون
٢٠٢٦م /١٤٤٨هـ

Proof: From the processing time and release times condition, get $C_{ð_g} = gp + r$, however, $d_{ð_g} = gp + r, \forall g$, which leads to $C_{ð_g} = d_{ð_g}$, it shows that every job is just-in-time (JIT), i.e., $T_{max} = V_{max} = 0$, then $(V_{max}, T_{max}, F_{max}, \sum F_g)$ reduced to $(F_{max}, \sum F_g) = (mp, \sum_{g=1}^{m} gp)$. Therefore, the ð schedule gives an efficient solution to the $\widetilde{G1}$ problem.

Case III: If $p_g = p, gp = d_g, \forall g$, then the ð = SRD schedule gives an (EF.S) for the $\widetilde{G1}$ problem.

Proof: for all g, $d_g = gp$, in the SRD, hence $T_{max} = \sum_{g=1}^{m} I_g = V_{max}$. Now, since $p_g = p$ for all g in the SRD, then

$$\sum_{g=1}^{m} C_g = \sum_{g=1}^{m} \sum_{i=1}^{g} I_i + \left(\frac{m^2+m}{2}\right) p.$$

(25)

But $\left(\frac{m^2+m}{2}\right) p$ is constant, and $F_{ð_g} = -r_{ð_g} + C_{ð_g}$, hence

$$\sum_{g=1}^{m} F_{ð_g} = -\sum_{g=1}^{m} r_{ð_g} + \sum_{g=1}^{m} C_{ð_g} = -\sum_{g=1}^{m} r_{ð_g} + \sum_{g=1}^{m} \sum_{i=1}^{g} I_i + \left(\frac{m^2+m}{2}\right) p \quad (26)$$

and $F_{max} = -r_{max} + C_{max} = -r_{max} + mp + \sum_{g=1}^{m} I_g$

(27)

Then problem $1//(V_{max}, T_{max}, F_{max}, \sum F_g) = \left(\sum_{g=1}^{m} I_g, \sum_{g=1}^{m} I_g, -r_{max} + mp + \sum_{g=1}^{m} I_g, -\sum_{g=1}^{m} r_{ð_g} + \sum_{g=1}^{m} \sum_{i=1}^{g} I_i + \left(\frac{m^2+m}{2}\right) p\right)$. Since $p_g = p$ for all g, then mp is constant, as is F_{max} in the equation (13). Therefore, the SRD schedule gives an (EF.S) for the $\widetilde{G1}$ problem.

Case IV: If $C_g = d_g, r_i \leq r_g$, and $p_i \leq p_g$, $\forall i, g = 1{:}m$ in a schedule ð [where ð = SPT = SRD], after that ð is an (EF.S) for the $\widetilde{G1}$ problem.

Proof: Due to the fact that $C_{ð_g} = d_{ð_g}, \forall g$(this implies that no early nor late jobs exist, hence $T_{ð_g} = V_{ð_g} = T_{max} = V_{max} = 0, \forall g$). Then the problem $\widetilde{G1}$ is reduced to $1/r_g/(F_{max}, \sum F_g)$, yet such a problem is solved in the SPT(SRD) rule. Hence, the schedule ð gives an (EF.S) for the $\widetilde{G1}$ problem.

Case V: If $r_g = r$ and there is a schedule ð ordered for its jobs based on the SPT rule, and for all $g \in ð, g$ is an early job, then ð is an (EF.S) for the $\widetilde{G1}$

problem. If $r_g = r$ and its jobs are scheduled and organized in accordance with the $ð = SPT$ rule, and for every $g \in ð, g$ is an early job, then ð is a (EF.S) for the $\breve{G}1$ problem.

Proof: Since g is an early job for all $g \in ð$ then $T_{ð_g} = V_{ð_g} = T_{max} = V_{max} = 0$, and the problem depends on $\sum F_g$, but $r_g = r$. Then the problem $1//(V_{max}, T_{max}, F_{max}, \sum F_g)$ reduced to $1/r_g/(F_{max}, \sum F_g)$. Implies that the SPT rule gives an (EF.S) for $\breve{G}1$ problem.

Case VI: If $C_g = d_g, \forall g$ in a table ð and pre-emption is allowed, then ð is the (EF.S) to the $\breve{G}1$ problem.

Proof: For all g, $C_{ð_g} = d_{ð_g}$. Hence $T_{ð_g} = V_{ð_g} = T_{max} = V_{max} = 0, \forall g$. Then the $\breve{G}1$ problem is reduced to $1/r_g/(F_{max}, \sum F_g)$, yet such a problem is solved in the SRPT rule. Hence, the schedule ð gives an (EF.S) for the $\breve{G}1$ problem.

Case VII: If there is a common due date for all tasks (i.e., $d_g = d$; $\forall g = 1{:}m$), and if the SRD gives $d \geq C_{max}$, then the SRD is the (EF.S) to the $\breve{G}1$ problem.

Proof: For any task, there will be no delay because $d \geq C_{max}$, i.e., hence $T_{ð_g} = V_{ð_g} = T_{max} = V_{max} = 0, \forall g$. Then, the $\breve{G}1$ problem is reduced to $1/r_g/(F_{max}, \sum F_g)$, and solved by SRD. Then the SRD gives the (EF.S) for the $\breve{G}1$ problem.

By evaluating the objective function Z_I, Table ١ provides illustrative examples of special cases of the ($\breve{G}1$) problem for $n = 6$.

Table 1. Illustrative examples of special cases of the ($\breve{G}1$) problem.

	r_g, p_g, and d_g	Conditions	Z_I

JOBS مجلة العلوم الأساسية
Journal of Basic Science
Print -ISSN 2306-5249
Online-ISSN 2791-3279
العدد الثامن والاربعون
٢٠٢٦م /١٤٤٨هـ

Case I	$r_k = 5, p_k = 12$ and $d_k = 90$.	$r_g = r, p_g = p$ & $d_g = d, \forall g = 1{:}m$	(0,0,72,252)
Case II	$r_k = 7, p_k = 9$ and $d_k =$ 16,25,34,43,52,61.	$p_g = p, r_g = r, \forall g$, and $d_g = gp + r$	(0,0,54,189)
Case III	$r_k =$ 4,6,8,9,4,3, $p_k =$ 12 and $d_k =$ 12,24,36,48,60,72.	$p_g = p, gp = d_g, \forall g$	(12,27,66,236)
Case IV	$r_j =$ 1414,14,8,5,0, $p_j =$ 20,12,10,8,6,5 and $d_j =$ 61,41,29,19,11,5.	$C_g = d_g, r_i \le r_g$, and $p_i \le p_g, \forall i, g = 1{:}m$	(0,0,47,111)
Case V	$r_k = r = 4, p_k =$ 10,10,12,14,16,20 and $d_k =$ 14,24,36,50,60,86.	$r_g = r$, $\forall i, g = 1{:}m$	(0,0,86,276)
Case VI	$r_k =$ 2,4,3,1,4,5, $p_k =$ 5,6,8,8,10,12 and $d_k =$ 7,13,21,29,39,41.	$C_g = d_g, \forall g$	(0,0,41,131)
Case VII	$r_k =$ 2,4,3,1,4,5, $p_k =$ 5,6,8,8,10,12 and $d_k = 60$.	$d_g = d;\ \forall g = 1{:}m,\ d \ge C_{max}$	(0,0,41, 131)

4. SPECIAL CASES FOR PROBLEMS $\breve{G}2$

Case I: If $r_g = r, p_g = p$ & $d_g = d\ \forall g = 1{:}m$, then any schedule provides an (OP.S) for the $\breve{G}2$ problem.

Proof: The same proof applies as in the first case, and therefore, we arrive at the following:

- if $C_{ð_g} < d$, then $V_{max} + T_{max} + F_{max} + \sum F_g = F_{max} + \sum F_g = mp + \sum_{g=1}^{m} gp = \left(\frac{m^2+3m}{2}\right)p$.
- if $d \le C_{ð_g} \le d + p, \forall g$, then $V_{max} + T_{max} + F_{max} + \sum F_g = \sum_{g=1}^{m} gp + mp + r + mp - d + r + mp - d$ $= \left(\frac{m^2+m}{2}\right)p + 3mp + 2r - 2d = \left(\frac{m^2+6m}{2}\right)p + 2r - 2d$.
- if $C_{ð_g} > d + p, \forall g$, hence $V_{max} + T_{max} + F_{max} + \sum F_g = p + p + mp + \sum_{g=1}^{m} gp = \left(\frac{m^2+2m+2}{2}\right)p$.

So, any of the schedules is an (OP.S) for the $\breve{G}2$ problem (since the five quantities are constant).

Case II: If $p_g = p, r_g = r, \forall g$ then any schedule ð fulfils $d_g = gp + r$ gives an (OP.S) for the $\breve{G}2$ problem.

Proof: The same proof applies as in the second case, and therefore, we arrive at the following:

$$V_{max} + T_{max} + F_{max} + \sum F_g = F_{max} + \sum F_g = mp + \sum_{g=1}^{m} gp = \left(\frac{m^2+3m}{2}\right) p.$$

Therefore, the ð schedule gives an (OP.S) for the $\breve{G}2$ problem.

Case III: If $gp = d_g, p_g = p, \forall g = 1{:}m$, then the schedule ð = SDR gives an (OP.S) for the $\breve{G}2$ problem.

Proof: The same proof applies as in the third case, and therefore, we arrive at the following:

$V_{max} + T_{max} + F_{max} + \sum F_g = \left(\sum_{g=1}^{m} I_g\right) + \left(\sum_{g=1}^{m} I_g\right) + \left(-r_{max} + mp + \sum_{g=1}^{m} I_g\right) + \left(-\sum_{g=1}^{m} r_{\eth_g} + \sum_{g=1}^{m}\sum_{i=1}^{g} I_i + \left(\frac{m^2+m}{2}\right) p\right)$. Since $p_g = p$ for all g, then $F_{max} = mp + \sum_{g=1}^{m} I_m - r_{max}$, but np is constant. Hence, the SRD schedule gives an (OP.S) for the $\breve{G}2$ problem.

Case IV: If $p_i \le p_g, r_i \le r_g$ and $C_g = d_g$, $\forall i, g = 1{:}m$ in a schedule ð [where ð = SPT = SRD], then ð is an (OP.S) for the $\breve{G}2$ problem.

Proof: The same proof applies as in the fourth case.

Case V: If $r_g = r$ and its jobs are scheduled and organised in accordance with the ð = SPT rule, and for every $g \in ð, g$ is an early job, then ð is an (OP.S) for the $\breve{G}2$ problem.

Proof: The same proof applies as in the fifth case.

Case VI: If $C_g = d_g, \forall g$ in a table ð and pre-emption are allowed, then ð is the (OP.S) to the $\breve{G}2$ problem.

Proof: The same proof applies as in the sixth case.

Case VII: If all jobs have one due date (i.e., $d_g = d;\ \forall g = 1{:}m$), and if the SRD gives $d \ge C_{max}$, then the SRD is the (OP.S) to the $\breve{G}2$ problem.

Proof: The same proof applies as in the seventh case.

The theorem 1 describes the relation between $1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)$ and $1/r_g/V_{max} + T_{max} + F_{max} + \sum F_g$ and shows that any (OP.S) to $1/r_g/V_{max} + T_{max} + F_{max} + \sum F_g$ problem is one of (EF.S) to the $1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)$ problem., The following theorem explains the relation between the four criterion functions and the four objective functions:

Theorem 1: The optimal solution for the $1/r_g/V_{max} + T_{max} + F_{max} + \sum F_g$ problem is one of the efficient solutions to the $1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)$ problem.

Proof: First, assuming $\breve{\partial}$ be a schedule that gives the (OP.S) to the problem $1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)$, assume that $\breve{\partial}$ doesn't provide an efficient solution to the $1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)$ problem, after that, there's an efficient schedule, for example, ρ for the $1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)$ problem where:

$V_{max}(\rho) \leq V_{max}(\breve{\partial}), T_{max}(\rho) \leq T_{max}(\breve{\partial}), F_{max}(\rho) \leq F_{max}(\breve{\partial})$, and $\sum F_g(\rho) \leq \sum F_g(\breve{\partial})$.

At least one where there is strict inequality, which means:

$V_{max}(\rho) + T_{max}(\rho) + F_{max}(\rho) \leq V_{max}(\breve{\partial}) + T_{max}(\breve{\partial}) + F_{max}(\breve{\partial}) + \sum F_g(\breve{\partial})$, After that ρ is a schedule that provides optimal solutions for problem $1/r_g/V_{max} + T_{max} + F_{max} + \sum F_g$, yet $\breve{\partial}$ is an optimal schedule, and that contradicts the assumption in this study, then $\breve{\partial}$ should provide an efficient solution to the $1/r_g/(V_{max}, T_{max}, F_{max}, \sum F_g)$ problem.

By evaluating the objective function Z_{II}, Table ٢ provides illustrative examples of special cases of the ($\widetilde{G2}$) problem for $n = 6$.

JOBS مجلة العلوم الأساسية
Journal of Basic Science
Print -ISSN 2306-5249
Online-ISSN 2791-3279
العدد الثامن والاربعون
٢٠٢٦م /١٤٤٨هـ

Table 2. Illustrative examples of special cases of the $(\breve{G2})$ problem.

	r_g, p_g, and d_g	Conditions	Z_{II}
Case I	$r_k = 5, p_k = 12$ and $d_k = 90.$	$r_g = r, p_g = p$ & $d_g = d, \forall g = 1{:}m$	324
Case II	$r_k = 7, p_k = 9$ and $d_k =$ 16,25,34,43,52,61.	$p_g = p, r_g = r, \forall g,$ $d_g = gp + r$	243
Case III	$r_k =$ 4,6,8,9,4,3, $p_k =$ 12 and $d_k =$ 12,24,36,48,60,72.	$p_g = p, gp = d_g, \forall g$	341
Case IV	$r_j =$ 1414,14,8,5,0, $p_j =$ 20,12,10,8,6,5 and $d_j =$ 61,41,29,19,11,5.	$C_g = d_g, r_i \leq r_g$, and $p_i \leq p_g, \forall i, g = 1{:}m$	158
Case V	$r_k = r = 4, p_k =$ 10,10,12,14,16,20 and $d_k =$ 14,24,36,50,60,86.	$r_g = r$, $\forall i, g = 1{:}m$	362
Case VI	$r_k =$ 2,4,3,1,4,5, $p_k =$ 5,6,8,8,10,12 and $d_k =$ 7,13,21,29,39,41.	$C_g = d_g, \forall g$	172
Case VII	$r_k =$ 2,4,3,1,4,5, $p_k =$ 5,6,8,8,10,12 and $d_k = 60.$	$d_g = d;\ \forall g = 1{:}m,\ d \geq C_{max}$	172

5 .RESULTS and CONCLUSIONS

In this research, we arrived at the following conclusions based on the facts presented in points 3 and 4 .

- The optimal solution for problem 1/r_g/V_max+ T_max+F_max+∑F_g if and only if is one of the efficient solutions to the problem1\/r_g \/(V_max,T_max,F_max,∑F_g.(
- Any schedule provides an efficient (optimal) solution for problems (G1)˘ ((G2)˘) if d_g=d ,r_g=r,and p_g=p ∀g(g=1:m.(
- The SDR schedules ð provide efficient (optimal) solutions for problems (G1)˘ ((G2)˘), in the case of the fulfillment of one of the subsequent requirements:

1)If $gp = d_g, p_g = p, \forall g = 1{:}m,$

2) If $C_g = d_g, r_i \leq r_g$, and $p_i \leq p_g, \forall i, g = 1{:}m,$

JOBS
مجلة العلوم الأساسية
Journal of Basic Science
Print -ISSN 2306-5249
Online-ISSN 2791-3279
العدد الثامن والاربعون
٢٠٢٦م /١٤٤٨هـ

3) If $d_g = d$, and $d \geq C_{max}; \forall g = 1{:}m$..

6. Future Works

We can recommend research paths for further study: extending these special cases to parallel machines or flow shops and adding sequence-dependent setup times.

1. $P_m/r_k/(V_{max}, T_{max}, F_{max}, \sum F_g)$,
2. $P_m/r_k/V_{max} + F_{max} + T_{max} + \sum F_g$,
3. $F_m/r_k/(V_{max}, T_{max}, F_{max}, \sum F_g)$,
4. $F_m/r_k/V_{max} + F_{max} + T_{max} + \sum F_g$,
5. $P_m/S_k/(V_{max}, T_{max}, F_{max}, \sum F_g)$,
6. $P_m/S_k/V_{max} + F_{max} + T_{max} + \sum F_g$,
7. $F_m/S_k/(V_{max}, T_{max}, F_{max}, \sum F_g)$,

$F_m/S_k/V_{max} + F_{max} + T_{max} + \sum F_g$.

ACKNOWLEDGMENTS

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

REFERENCES

1. Agin, N. (1966). Optimum Seeking with Branch and Bound. Management Science, 13(4), B176–B185. http://www.jstor.org/stable/2627549
2. Ahmadi, R. H. (1990). Lower Bounds for Single-Machine Scheduling Problems. Naval Research Logistics, 37, 967–979.
3. Al-Zuwaini, M.K. (2011). One Machine Scheduling Problem With Release Dates and Two Criteria. Journal of Thi-Qar University2, 6(2), 1–14.
4. Allahverdi, A., Ng, C. T., Cheng, T. C. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. European Journal of Operational Research, 187(3), 985–1032. https://doi.org/10.1016/j.ejor.2006.06.060
5. Allaoua, H., & Brahim, B. (2017). New Properties for Solving the Single-Machine Scheduling Problem with Early/Tardy Jobs. Journal of Intelligent Systems, 26(3), 531–543. https://doi.org/10.1515/jisys-2016-0063

6. Baker, K. R. (2009). Principles of Sequencing and Scheduling (Second Edi). Wiley Series in Operations Research and Management Science. https://doi.org/10.1002/9781119262602

7. Batsyn, M. V, Kalyagin, V. A., & Pardalos, P. M. (2014). Models, Algorithms and Technologies for Network Analysis. In Springer Proceedings in Mathematics & Statistics (Vol. 104). Springer Proceedings in Mathematics & Statistics. https://doi.org/10.1007/978-3-319-09758-9

8. Chachan, H. A., & Jaafar, H. A. (2020). Exact Solutions for Minimizing Cost Function with Five Criteria and Release Dates on Single Machine. Ibn AL- Haitham Journal For Pure and Applied Sciences, 33(3), 140–157. https://doi.org/10.30526/33.3.2479

9. Cheachan, H. A., Ibrhim, M. H., & Mohammed, H. A. A. (2019). Scheduling of single machine with release dates to minimize multiple objective functions. IOP Conference Series: Materials Science and Engineering, 571(1). https://doi.org/10.1088/1757-899X/571/1/012003

10. Chen, R., Yuan, J., Ng, C. T., Yuan, J., & Ng, C. T. (2020). Single-machine hierarchical scheduling with release dates and preemption to minimize the total completion time and a regular criterion. European Journal of Operational Research, 293(1), 79–92. https://doi.org/10.1016/j.ejor.2020.12.006

11. Chu, C., Sagep, P., & Metz, T. (2000). Efficient heuristics to minimize total flow time with release dates. Operations Research Letters, 12(November 1992), 321–330. https://dl.acm.org/doi/10.1016/0167-6377(92)90092-H

12. Daowd, O. osama, Chachan, H. A., & Daway, H. G. (2022). Solving Multi-Objectives Function with Release Dates Problem using Branch and Bound Methods. Journal of Al-Qadisiyah for Computer Science and Mathematics, 14(2), 1–21. https://doi.org/10.29304/jqcm.2022.14.2.948

13. de Weerdt etal. (2021). Single-machine scheduling with release times, deadlines, setup times, and rejection. European Journal of Operational Research, 291(2), 629–639. https://doi.org/10.1016/j.ejor.2020.09.042

14.E . L . Lawler. (2013). Optimal Sequencing of a Single Machine Subject to Precedence Constraints. Management Science Theory Series (Jan ., 1973), 19(5), 544–546.

15.Grigoreva, N., & Didenko, A. (2025). Approximation Algorithms for Single Machine Scheduling Problem with Delayed Precedence Constraints. CSIT Conference 2025, 248–251. https://doi.org/https://doi.org/10.51408/csit2025_60

16.Hoogeveen, H. (2005). Multicriteria scheduling. European Journal of Operation Research, 167(3), 592–623. https://doi.org/https://doi.org/10.1016/j.ejor.2004.07.011 Get rights and content

17.Johnson, S. M. (1954). Optimal two- and three- stage production schedules with setup times included. Naval Research Logistics Quarterly, I(1), 61-68.

18.Lakshminarayan, S., Lakshmanan, R., Papineau, R. L., & Rochette, R. (1978). Optimal Single-machine Scheduling with Earliness and Tardiness Penalties. Operations Research, 26(6), 1079–1082. http://www.jstor.org/stable/170267.

19.Lee and Liman. (1992). Single machine flow-time scheduling with scheduled maintenance. Acta Informatica- Springer-Verlag, 382, 375–382.

20.Michael L. Pinedo. (1994). Scheduling Theory, algorithms, and systems. In Springer Science+Business Media (Fifth Edit). https://doi.org/10.1007/978-3-319-26580-3

21.Nagham 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

22.Nagham M. Neamah and Bayda A. Kalaf. (2024b). Minimizing Total Completion Time, Total Earliness Time, and Maximum Tardiness for a Single Machine scheduling problem. Ibn AL-Haitham Journal For Pure and Applied Sciences, 37(1), 386–402. https://doi.org/10.30526/37.1.3094

23.Nagham M. Neamah and Bayda A. Kalaf. (2024c). 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

24.Nagham M. Neamah and Bayda A. Kalaf. (2024d). Solving Tri-criteria: Total Completion Time, Total Earliness, and Maximum Tardiness Using Exact and Heuristic Methods on Single-Machine Scheduling Problems. Mathematical Modelling of Engineering Problems, 11(4), 987–995. https://doi.org/10.18280/mmep.110415

25.Nagham 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/https://doi.org/10.52866/ijcsm.2024.05.03.002

26.Opérationnelle, R., & Opérationnelle, R. E. (1996). A new class of scheduling criteria and theid optimization. Recherche Opérationnelle/Opérations Research, 30(2), 171–189. https://www.numdam.org/item/?id=RO_1996__30_2_171_0

27.Osama, O., Ali, H., & Daway, H. G. (2022). Theoretical approaches and special cases for a single machine with release dates to minimize four criterion. 13(1), 2075–2085.

28.Phrueksanant, J. (2013). Machine scheduling using the Bees algorithm [Cardiff University - United Kingdom]. http://orca.cf.ac.uk/58594/

29.Pinedo, M. (1992). Scheduling : Theory, Algorithms and System Dvelopment. Operations Research Proceedings 1991, March, 35–42. https://doi.org/10.1007/978-3-642-46773-8_5

30.Smith, W. E. (1956). Various optimizers for single-stage production. Navel Research Logistics Quarterly, 3, 59–66. https://doi.org/10.1002/nav.3800030106

31.Stevenson. (2015). Operations Management (W. j . Stevenson (ed.); Thirteenth). McGraw-Hill Education. https://doi.org/LCC TS155 .S7824 2018 | DDC 658.5--dc23

32.Taylor, P., Mahnam, M., & Moslehi, G. (2009). A branch-and-bound algorithm for minimizing the sum of maximum earliness and tardiness with unequal release times. Taylor & Francis, December 2014, 37–41. https://doi.org/10.1080/03052150802657290

33.Woeginger, G. J. (1999). Approximability and Nonapproximability Results for Minimizing Total Flow Time on a Single Machine. Eindhoven University of Technology, 28(4). https://doi.org/https://doi.org/10.1137/S0097539796305778

مجلة العلوم الاساسية
للعلوم التربوية والنفسية وطرائق التدريس للعلوم الاساسية

مجلة العلوم الاساسية
للعلوم التربوية والنفسية وطرائق التدريس للعلوم الاساسية