Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 1s (2025)

Solving Tricriteria Machine Scheduling Problem using New Efficient Method

Yasameen M.Mohammed[1], Iraq T.Abbas[2]
[1]Department of Mathematics and Computer Applications, College of Science, Al-Nahrain University, Baghdad, Iraq.
[2]Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq.
Corresponding author : Yasameen M.Mohammed
Yasmin.Mohammed1103@sc.uobaghdad.edu.iq[1] , Iraq.t@sc.uobaghdad.edu.iq[2]

Article History:

Received: *25-07-2024*

Revised: *20-09-2024*

Accepted: *07-10-2024*

Abstract

In this study, an MBA model which can select a single machine for production and simultaneously consider many criteria is proposed. In this paper we present the description of the machine scheduling problem (MSP), which is the set of n tasks accomplished by a single machine. The task will be to minimise a function with reference to the criteria that are enumerated below: ∑C_j – total time whereas R_L – range of lateness; T_max – maximum of tardiness. Proofs have been made to conclude that this problem is NP hard. The matter we are concerned with calls for the emergence of a number of subissues, all of which will be discussed in more detail later on. In the theoretical part of our investigation, we have shown that the SPT rule offers a viable solution to the issue that we are attempting to solve. Furthermore, we have shown that it is possible to implement certain manifestations of the dominance rule (DR). In order to solve the suggested MSP tricriteria, the Branch and Bound (BAB) algorithm is used throughout the practical portion of the evaluation process. The objective of this approach is to identify a set of solutions that are not only successful but also efficient for 1//(∑C_j,R_L,T_max) up to n=18 jobs. In addition, the BAB method is used in conjunction with DR for a maximum of n=39 tasks within a reasonable amount of time in order to uncover estimated effective solutions for the problem at hand.

Keywords: Simulated machine issue, Multiple criteria, overall completion duration, maximum delay, Latency variety, Branch and Bound.

1. Introduction

Scheduling difficulties belong to the broad category of problems referred to as combinational optimization. It can be defined as an orderly approach to choosing among different alternatives often used in many manufacturing and service industries. Resource allocation is the way of assigning resources to tasks and time periods also the goal of optimising one or more criteria in addition (Pinedo, 2008 [1]). Scheduling theory has been a significant research focus in several practical domains such as manufacturing systems, computer science, industrial management, transportation, agriculture, hospitals, and others (Agin, 1996[2]). Resources and tasks are referred to as machine jobs, respectively.

Scheduling means the assignment of scarce resources in excess to a definite number of operations, besides in a given period of time.e. Assets include different things like apparatus in a workshop, tracts at an airport, workers at a construction site, and processing units in a computing system. Employment includes matters under the area of industry, takeoff and descents of aeroplanes, sequences of construction projects, and computation of computer programming. The purpose of scheduling is to allocate human resources for the job with the purpose of achieving on or several goals with the best results. There are myriad sorts of problem classes in manufacturing scheduling. Processing facilities can be categorised into single-chain machine, simultaneous-chain machine, flow-chain machine, and job-chain machine [4].

In recent years, there has been a significant amount of interest in the multicriteria scheduling issue. Nagar et al. (1995) have produced a detailed survey of multicriteria models throughout this time. A effective resolution of two different kinds of problems is shown by the results. The first one covers the difficulties associated with minimising the order of criteria in a lexicographical style. Several examples of hierarchical minimisation difficulties may be found in the research carried out by Smith (2), Nagar et al. (1995) (5), and Hoogeveen (2005) (7). Additionally, the concurrent technique incorporates two distinct approaches. The first type will calculate all of the possible schedules and then choose the one that has the best aggregate function goals value. This will be done by combining each of the criteria. Calculating the absolute worth of such goals is the task that remains to be done. The simultaneous minimisation of a variety of goal functions is a method that may be used to solve a great deal of scheduling challenges simultaneously. The article by Hoogeveen (2005) [7] provides a complete analysis of the most important discoveries on multicriteria scheduling. Van Wassenhove and Gelders (1980) [8] were the ones that launched the first inquiry in the subject of simultaneous computing to be conducted. Researchers evaluated the effectiveness of overall completion times and maximum tardiness in relation to a single machine challenge by putting them through a series of tests. Multiple criteria are discussed in further depth in the references. (9, 10, 11).

During the duration of this investigation, we investigate the challenge of minimising a certain set of multicriteria while simultaneously scheduling n tasks on a single computer. There is only one computer that can handle all of the n tasks at the same time, thus each and every one of them can only be handled by that one machine. In task j, the processing time and the due date are connected to one another. At approximately instance zero, all of the works are available to be examined at the same time concurrently. Obtaining a collection of Pareto-optimal solutions to the problem of 1//F (∑C_j,R_L,T_max) has been accomplished. In the second part, we will investigate the mathematical formulation of the issue that involves the interval 1//P∑C_j,R_L,T_max. A new upper and lower limit for the BAB will be suggested in the third portion of the research paper. There are two heuristic techniques that are provided in the fourth part in order to find a solution that is close to optimum for the issue that has been specified. In the fifth part, the results, which include both a conceptual analysis and a comparison, are described in depth. Further, the most significant findings and a few suggestions will be offered in the sixth paragraph, which will also include some recommendations.

1.1 Important Notations

This report makes use of a few indications:

n	:	Number of jobs.
p_j	:	Processing time of jobs j.
d_j	:	Due date of jobs j.
C_j	:	Completion time of job j, where $C_j = \sum_{k=1}^{j} p_k$.
$\sum C_j$	:	Total completion time.
L_j	:	Lateness of job j, $L_j = C_j - d_j$.
R_L	:	Range of lateness, $R_L = L_{max} - L_{min}$.
T_j	:	Tardiness of job j, $T_j = max\{L_j, 0\}$.
T_{max}	:	Maximum Tardiness of all jobs, $T_{max} = max\{T_j\}$.
DR	:	Dominance Rules
WDR	:	In addition, out DR.

1.2 Machine Scheduling Problem

In this paper we need some basic definitions.

Definition (1) [12] and [15]: Assuming that there is a scheduling problem P, any schedule $\sigma \in S$, that fulfils the requirements of that issue P is considered viable, wherein S is the collection of all programs.

Definition (2) [7] and [16]**:** When there is no reasonable schedule π that meets simultaneously $f(\pi) \leq f(\sigma)$and $g(\pi) \leq g(\sigma)$, wherein no less than both of the inequality is rigorous, then a viable schedule σ is Pareto optimum, or non-dominated (efficient) In addition regard to the achievement criteria f as well as g.

Definition (3): (Shortest Processing Time (SPT) rule) [4] and [17]**:** Jobs are sequenced in non-decreasing order of processing times (p_j), (i.e. $p_1 \leq p_2 \leq \cdots \leq p_n$). This set of rules was applied to resolve the issue $1//\sum C_j$.

Definition (4): Earliest Due Date (EDD) rule [7]: The descending sequence of operations is based on their due date (d_j) and is not declining, meaning that (i.e. $d_1 \leq d_2 \leq \cdots \leq d_n$). The problem $1//T_{max}$ is minimized by applying the preceding rule.

Definition (5) [13]: In a multi-criteria settlement getting challenge, the word "**optimizes**" refers to a solution regarding which it is impossible to develop or improve every goal devoid leaving any additional objective inferior.

Definition (6) [14]**:** Assuming we have been unable to locate a different schedule S' that satisfies $f_j(S') \leq f_j(S), j = 1, \ldots, k$ and over and above all among the aforementioned holds as a rigorous inequality, then schedule S is considered economical. There is a different way to say that S'dominates S.

2. Description of Tricriteria Scheduling$1/(\sum C_j, E_{max}, R_L)$ Problem

Let $N = \{1,2, \ldots, n\}$,be the set of jobs which are available at time zero to be scheduled on a single machine. Each job$j \in N$, has positive integer processing time$p_i$and positive integer due date d_j. The machine can handle only one job at a time using the proposed three-field categorization by Graham et al [3], the MSP denoted by $1//F(\sum C_j, E_{max}, R_L)$. The goal is to endeavor to identify the collection of effective approaches for the machine that can be expressed in the following manner for a particular timetable $S = (1,2, \ldots, n)$:

$$
\left.
\begin{aligned}
& Min \left\{\sum C_j, R_{L(S)}, T_{max}\right\} \\
& \text{Subject to} \\
& C_1 \geq p_{S(1)}, \\
& C_j = C_{(j-1)} + p_{S(j)}, \qquad j = 2,3, \ldots, n \\
& L_j = C_j - d_{S(j)}, \qquad j = 1,3, \ldots, n. \\
& T_j \geq C_j - d_{S(j)}, \qquad j = 1,3, \ldots, n. \\
& R_L(S) = L_{max}(S) - L_{min}(S), \\
& T_j \geq 0, \qquad j = 1,3, \ldots, n.
\end{aligned}
\right\} \quad \ldots(P)
$$

This problem (P) is difficult to solve and find the set of all efficient solutions.

3. Efficient Solutions for *P*-Problem using Branch and Bound Algorithm

In this section, we propose two techniques; classical Branch and Bound (BAB) or we can say BAB In addition out DR (WDR) to ascertain a collection of the best alternatives for issue (P). The phases of the BAB(WDR) are listed below:

BAB(WSR) Algorithm

Step (1): **INPUT** n, $p_i$and $d_j$for $j = 1,3, \dots, n$.

Step (2): **SET**S= ϕ, define$F(\sigma) = \left(\sum C_{\sigma(j)}, R_L(\sigma), T_{\max}(\sigma)\right)$, for any σ.

Step (3): Find the upper bound UB by σ=SPT rule. For this order σ, compute $F(\sigma), j = 1,2, \dots, n$. And set the upper bound$UB = F(\sigma)$at the parent node of the search tree.

Step (4): Every node in the searching tree of the Branch and Bound method, compute the lowest bound LB(δ) for each incomplete job sequence δ as follows: LB (δ) = cost of sequenced jobs (δ) for achieving the goal functionalities + cost of sequenced tasks derived by using the SPT technique.

Step (5): Branch form all nodes where the lower bound is less than or equal to the upper bound (LB ≤ UB).

Step (6): At the final stage of the BAB technique, we receive a set of solutions when $F(\delta)$ is used to indicate that what happens should be included to the set S. If it isn't overwhelmed by the efficient alternatives that were originally identified in S, subsequently Filtering S is applied.

Step (7): **STOP**.

In the best case scenario, the time taken to solve the Problem (P) is moderately reasonable in the BAB(WDR) up to the number n=17. In this part, we also present another BAB which relies on the DR. This kind of BAB is called BAB(DR) and the aim here is to reduce the number of opened nodes thus taking lesser time and increasing number of n for the issues solved. It has some resemblance to BAB(WDR) method, although the primary phases of the approach is similar too there are several methods that are unique to this approach. The following is a list of the stages that are included in the BAB(DR):

BAB(DR) Algorithm

Step (1): **INPUT** n, $p_i$and $d_j$for $j = 1,2, \dots, n$. Find Adjacency Matrix A.

Step (2): **SET**S= ϕ, define $F(\sigma) = \left(\sum C_{\sigma(j)}, R_L(\sigma), T_{\max}(\sigma)\right)$, for any σ.

Step (3): Find the upper bound UB by σ= SPT rule. For this order σ, compute$F(\sigma), j = 1,2, \dots, n$. And set the upper bound$UB = F(\sigma)$at the parent node of the search tree.

Step (4): In addition, every node in the BAB method's searched tree and every partial collection of jobs δ, compute a lower bound LB (δ) in the following manner: Cost of subsequence jobs acquired through sequencing the jobs in the SPT method + cost of sequenced jobs (δ) considering the goal functionalities equals LB (δ).

Step (5): Branch from each node in addition LB ≤ UB and $i \to j$.

Step (6): At the final stage of the BAB technique, we receive a set of solutions when $F(\delta)$ is used to indicate that what happens should be included to the set S. If it isn't overwhelmed by the efficient alternatives that were originally identified in S, subsequently Filtering S is applied.

Step (7): **STOP**.

The BAB(DR) we solve problem (P) up to n=39 in a reasonable time.

4. Heuristic Method for *P*-problem

As the SPT method solves the $1//\sum C_j$ challenge, the first heuristic technique compute s the objective function, puts the second job in first place, make plans as the remaining jobs according to the SPT rule, compute s the target functioning, and so on before n orders are acquire ed. The following belong to SPT-EDD-SCRLT's primary processes:

Algorithm (3): SPT-EDD-SCRLT Heuristic Method.
Step (1): INPUT n, $p_j$and $d_j, j = 1,2, \dots, n,\ \delta = \emptyset$.
Step (2): Make plans jobs in SPT rule (σ_1), and compute $F_{11}(\sigma_1); \delta = \delta \cup \{F_{11}(\sigma_1)\}$.
Step (3): FOR i=2,…,n, put job i in the first place of $\sigma_{i-1}$to acquire σ_i and compute $F_{1i}(\sigma_i)$; $\delta = \delta \cup \{F_{1i}(\sigma_i)\}$.
END;
Step (4): Make plans jobs in EDD rule (π_1, compute $F_{21}(\pi_1)$; $\delta = \delta \cup \{F_{21}(\pi_1)\}$.
Step (5): FOR i=2,…,n, put job i in the first place of $\pi_{i-1}$to acquire π_i and compute $F_{2i}(\pi_i)$; $\delta = \delta \cup \{F_{2i}(\pi_i)\}$.

END;

Step (6): Filter set δ to acquire as a set of efficient solution of P-problem
Step (7): OUTPUT The set of efficient solutionδ.
Step (8): END.
The idea of the second heuristic method is summarized by finding a sequence sort in addition minimum p_j, d_j and d_j which is not contradiction in addition DR and compute the objective function, The main steps of DR-SERL Tares follows:
Algorithm (4): DR_SCRLT Heuristic Method.
Step (1): INPUT: n, p_j and$d_j, j = 1,2, \dots, n$.
Step (2): Apply preplaces)1) to find DR adjacency matrix A;
$\sigma = \emptyset, N = \{1,2, \dots, n\}$.
Compute $s_j = d_j - p_j, \forall j \in N, \delta = \emptyset$.
Step (3): Find a sequence σ_1 In addition minimum p_j which is not contradiction in addition DR (matrix A), if ∃ Over and above one job break tie capricious, $\delta = \delta \cup \{\sigma_1\}$.
Step (4): Find a sequence σ_2 In addition minimum d_j which is not contradiction in addition DR (matrix A), if ∃ Over and above one job break tie capricious, $\delta = \delta \cup \{\sigma_2\}$.
Step (5): Find the dominated sequence set δ' from δ.
Step (6): Compute $F(\delta)$.
Step (7): OUTPUT The set of efficient solutionδ.

Step (8): END.

5. Practical Result of P problems

The at unplanned values of p_j and d_j for all example are generated depending on the uniform allocation s.t. $p_j \in[1,10]$ and $d_j \in [1,70]$ under condition $d_j \geq p_j$, for j=1,..,n.

Before showing all the outcomes tables, we introduce some important abbreviations:

Ex	:	Example Number.
Av	:	Average.
NS	:	Number of efficient Solution.
ANS	:	Average number of efficient solutions.
T/S	:	CPU-Time per second.
AT/S	:	Average of CPU-Time per second.
MOF	:	Multi Objective Function.
OP	:	Optimal Value of P_1-problem.
R	:	0 < Real < 1.
F	:	Objective Function of P-problem.

The outcomes of applying CEM and BAB(WDR) which are compared in addition CEM for P-problem, n=3:10 are shown in table (1).

Table (1): Comparative analysis of BAB and CEM for P-problem, n=3:10.

n	**CEM**			**BAB(WDR)**		
	OP	**TIME**	**NES**	**MOF**	**TIME**	**NES**
	Av(F)	**AT/S**	**ANES**	**AMAE**	**AT/S**	**ANES**
3	(33.5,15.9,10.2)	R	2.4	(33.1,14.9,15.1)	R	3.2
4	(58.0,15.6,17.6)	R	6.6	(47.4,13.8,16.9)	R	4.8
5	(79.5,8.9,17.0)	R	6.4	(51.3,11.6,13.8)	R	7.2
6	(85.5,14.2,23.0)	R	13.4	(104.6,8.8,21.9)	R	6.4
7	(122.0,8.3,21.7)	R	17.2	(110.4,11.5,24.1)	R	10.4
8	(173.3,9.9,33.7)	1.1	13.2	(180.0,11.0,36.5)	R	13.6
9	(211.7,8.8,36.5)	10.6	20.0	(213.8,11.2,41.4)	R	9.0
10	(254.7,9.8,40.8)	109.8	15.2	(257.0,10.0,42.7)	R	12.0
AV	(127.275,11.425,25.0625)	15.1875	11.8	(124.7,11.6,26.55)	R	8.325

From table (1), we notice that BAB(WDR) is more accurate to CEM outcomes because it's found all the solutions for P-problems in addition no matter that the optimal schedule.

In Table (2), a comparison has been made between CEM and HUE1, HUE2 for P-problem for n=3:10.

Table (2): Comparative analysis of CEM and HUE1, HUE2 for P-problem, n=3:10.

n	**CEM**			**HUE1**			**HUE2**		
	OP	**TIME**	**NES**	**MOF**	**TIME**	**NES**	**MOF**	**TIME**	**NES**
	Av(F)	**AT/S**	**ANES**	**Av(F)**	**AT/S**	**ANES**	**Av(F)**	**AT/S**	**ANES**
3	(33.5,15.9,10.2)	R	2.4	(34.50,16.03,11.40)	R	2.20	(36.30,9.50,10.90)	R	1.40
4	(58.0,15.6,17.6)	R	6.6	(42.17,15.67,15.17)	R	3.20	(66.40,12.00,17.90)	R	2.00
5	(79.5,8.9,17.0)	R	6.4	(73.40,11.92,19.05)	R	4.80	(63.70,12.70,17.00)	R	2.00

6	(85.5,14.2,23.0)	R	13.4	(92.51,12.55,24.44)	R	4.80	(89.70,13.90,22.20)	R	2.00
7	(122.0,8.3,21.7)	R	17.2	(127.02,13.64,29.90)	R	5.80	(125.60,11.10,25.50)	R	2.00
8	(173.3,9.9,33.7)	1.1	13.2	(194.68,13.44,39.52)	R	6.60	(139.50,12.50,28.60)	R	2.00
9	(211.7,8.8,36.5)	10.6	20.0	(215.47,12.38,44.16)	R	7.00	(186.10,15.30,34.40)	R	2.00
10	(254.7,9.8,40.8)	109.8	15.2	(265.40,11.66,49.89)	R	7.20	(243.10,13.20,40.80)	R	2.00
AV	(127.275,11.425,25.0625)	15.1875	11.8	(130.64375,13.41125,29.19125)	R	5.2	(118.8,12.525,24.6625)	R	1.925

For *n*=3:10, we notice that HEU1 and HUE2 starts to give minimum values for *P*-problem compared in addition outcomes of CEM.

The comparison outcomes of HUE1, HUE2 In addition BAB, for *P*-problem, *n*=20,40(20): 200, appears in the table (3).

Table (3): Comparative analysis of HUE1, HUE2 In addition BAB, for *P*-problem, n=20:200.

n	**BAB**			**HUE1**			**HUE2**		
	OP	**TIME**	**NES**	**OP**	**TIME**	**NES**	**OP**	**TIME**	**NES**
	Av(F)	**AT/S**	**ANES**	**Av(F)**	**AT/S**	**ANES**	**Av(F)**	**AT/S**	**ANES**
20	(945.2,8.9,102.3)	R	16.4	(819.54,17.77,98.10)	R	10.40	(877.40,12.70,86.60)	R	2.00
40	(3431.6,12.1,214.9)	3.5	30.6	(3204.87,15.84,202.36)	R	13.00	(3523.30,10.80,193.50)	R	2.00
60	(7057.4,10.4,307.1)	8.4	20.0	(7427.28,16.22,321.93)	R	12.60	(8025.10,11.50,298.20)	R	2.00
80	(11872.3,9.5,399.9)	18.0	15.4	(12944.71,18.76,433.41)	R	13.00	(14450.20,8.80,417.60)	R	2.00
100	(19200.9,9.1,521.7)	34.0	14.8	(21148.27,15.32,551.29)	R	12.80	(22338.10,8.20,523.20	R	2.00
120	(27980.7,7.7,635.5)	48.8	12.8	(29292.47,15.08,649.73)	R	13.40	(32200.10,8.50,625.60)	R	2.00
140	(37641.3,7.1,734.9)	76.1	12.0	(40545.67,18.05,774.52)	R	12.80	(44762.30,5.90,751.00)	R	2.00
160	(49320.6,5.6,849.2)	97.2	10.2	(52514.47,15.72,879.55	R	12.40	(60281.30,7.70,874.20)	R	2.00
180	(63858.0,6.1,972.1)	185.2	10.2	(63008.89,17.17,960.08)	R	12.20	(73026.20,5.10,960.30)	1.11	2.00
200	(78689.6,5.3,1084.5)	218.9	10.4	(80882.31,19.09,1104.45)	R	13.00	(88268.90,4.40,1045.90)	1.33	2.00

Notice that the Heuristic HEU1 gives better outcomes from HUE2 compared in addition BAB for *P*-problem for *n*=4:10.

In table (4) we compare the outcomes acquire ed from heuristic HUE1 and HUE2 for *P*-problem, n=500,1000:(500):4500.

Table (4): Outcomes of comparison of HUE1 and HUE2 for *P*-problem, n=500,1000:(500):4500.

n	HUE1			HUE2		
	OP	TIME	NES	MOF	TIME	NES
	Av(F)	AT/S	ANES	Av(F)	AT/S	ANES
500	(493126.62,16.86,2728.22)	4.16	12.80	(559413.30,1.00,2745.60)	6.52	1.80
1000	(1987507.43,17.57,5520.41)	19.05	12.80	(1918036.00,0.00,5429.40)	30.71	1.00
1500	(4521076.82,17.91,8308.13)	49.90	12.20	(4326882.40,0.00,8195.20)	79.40	1.00
2000	(7911944.32,16.04,10978.33)	101.84	12.40	(7727904.20,0.00,10976.80)	164.93	1.00
2500	(12297995.08,15.48,13699.35)	181.22	12.40	(12155643.20,0.00,13818.40)	297.88	1.00
3000	(17929735.53,18.36,16568.15)	293.46	12.60	(17396506.80,0.00,16516.80)	488.12	1.00
3500	(24054146.56,15.79,19119.26)	444.77	12.60	(23675196.00,0.00,19261.60)	746.43	1.00
4000	(31691861.51,18.31,22001.70)	641.26	12.80	(30967259.20,0.00,22031.60)	1084.90	1.00
4500	(40150599.50,15.23,24772.67)	896.01	12.60	(39024255.40,0.00,24740.80)	1512.99	1.00
Av.	(15670888.1522,16.83888,13744.0244)	292.40777	12.5777	(15305677.3888,0.1111,13746.2444)	490.20888	1.0888

7. Conclusions and Future Works

In this study, two approaches to BAB are presented: the first one is addition algorithm and the second is the DR algorithm but instead of a subtraction it employs additions. Since BAB(WDR) is only constrained by the given criteria LB≤UB, a higher precision range is assumed for 4≤n≤18: higher NS values than BAB(DR). Nevertheless, though being faster and requiring significantly less CPU time than other methods, BAB(DR) is the method that yields the lowest accuracy of all. Each of the heuristic techniques; SPT-EDD-SCRLT and DR-SCRLT performed excellently, especially when introduced as the solution to the P-problem. For that reason, it is possible that many of them originated from the P-problem, which may include 1//(∑C_j+R_L+T_max) and 1//Lex∑C_j+R_L+T_max) and the way in which different possibilities are then utilised in an attempt to solve them can be observed and examined. Hence, in order to obtain efficient, approximative solutions to the aforementioned P-problem in cases when n values are more than 100, the author suggests to develop the further research by using more numerous local search methods. Among the strategies that are covered under this category of tactics are simulation of other techniques including anneal, particle swarm optimisation, the genetic algorithm and Bees algorithm. Other techniques that come under this category are Bees algorithms.

References

[1]. Pinedo M. L., "Scheduling: theory, algorithms and systems", New York: Springer (2008).
[2]. Agin N., "Optimal seeking In addition branch and bound ", Mgmt. Sci. 13, 176-185 (1996).
[3]. Jackson J. R., 'Scheduling a production line to minimize maximum tardiness", Research Report 43, Management Science Research Project, University of California, Los Angeles, USA (1955).
[4]. Allard D. M., "A multi-objective genetic algorithm to solve single machine scheduling problems using a fuzzy fitness function", M. Sc. Thesis, Dept. of Industrial Engineering, College of Engineering and Technology, Univ. of Ohio (2007).
[5]. Nagar A., Haddock J. and Heragu S., "Multiple and bi-criteria scheduling: A literature survey", European Journal of Operational Research 81, 88-104 (1995).
[6]. Smith W.E., "Various optimizers for single – stage production", Navel Research Logistics Quarterly, 3, 59-66, 1956.

[7]. Hoogeveen H., "Invited review of multicriteria scheduling", European Journal of Operational Research 167, 592-623 (2005).
[8]. Van Wassenhove L. N. and Gelders F., "Solving a bicriterion scheduling problem", European Journal of Operational Research 4, 42-48(1980).
[9]. Tariq, S. Abdul-Razaq and Sally A. Ali, 2016, "*A Comparison of Local Search Algorithm for Multicriteria Scheduling Problems*", M.Sc. thesis, University of Al-Mustansiriyah, College of Science, Dept. of Mathematics.
[10]. Faez H. Ali and Shrmeen B. Abdul-Kareem, 2017, "*Scheduling a Single Machine to Minimize Max Tardiness, Max Late Work and Total Late Work*". Mathematics and Statistics Journal, 3(1): 1-17.
[11]. Hanan A. Chachan and Alaa S. Hameed, 2019, "*Exact Methods for Solving Multi-Objective Problem on Single Machine Scheduling*", Iraqi Journal of Science, 2019, Vol. 60, No.8, pp: 1802-1813, DOI: 10.24996/ijs.2019.60.8.17.
[12]. French S., "Sequencing and scheduling: An introduction to the mathematical of the job shop", John Wiley and sons, New York (1982).
[13]. Faez H. Ali, 2015, "*Improving Exact and Local Search Algorithms for Solving Some Combinatorial Optimization Problems*", Ph. D., Thesis, MustansiriyahUniversity, College of Science, Dept. of Mathematics.
[14]. Hoogeveen, J.A., 1992,"*Single-machine bi-criteria scheduling*", Ph. D. Dissertation, Center for mathematics and Computer science, Amsterdam. The Netherlands.
[15]. Abbas, I. T., & Ghayyib, M. N. (2024). *Using Sensitivity Analysis in Linear Programming with Practical Physical Applications*. Iraqi Journal of Science, 65(2).
[16]. Sheah, R. H., & Abbas, I. T. (2021). *Using multi-objective bat algorithm for solving multi-objective non-linear programming problem*. Iraqi Journal of Science, 997-1015.
[17]. Abdul-Zahra, I., Abbas, I. T., Kalaf, B. A., Bakar, R. A., June, L. W., & Monsi, M. B. (2016). *The Role of Dynamic Programming in the Distribution of Investment Allocations between Production Lines with an Application*. International Journal of Pure and Applied Mathematics, 106(2), 365-380.