Journal of Al Rafidain University College **Issue No. 46/ 2020**
The (15th & the 2nd International) Conference of Statistical Applications **ISSN (1681- 6870)**

Efficient Algorithms to Solve Tricriteria Machine Scheduling Problem

Manal Ghassan Ahmed[1], Faez Hassan Ali[2]
[1,2] Mathematics Dept., College of Science, Mustansiriyah University, Baghdad, Iraq

Abstract

The multicriteria single machine model is presented in this paper. We consider the machine scheduling problem (MSP) of *n* jobs on a single machine minimize a function of tricriteria: total completion time ($\sum C_j$), range of lateness (R_L) and maximum tardiness (T_{max}) which is an NP-hard problem.

In the theoretical part of this work, we introduce the mathematical formulation of the discussed problem then demonstrate the importance of the dominance rule (DR) which can be applied in this problem to improve the good solutions. While in the practical part, one of the important exact methods; Branch and Bound (BAB) algorithm is applied to solve the suggested MSP tricriteria by finding a set of efficient solutions for $1//(\sum C_j, R_L, T_{max})$ up to *n*=18 jobs and BAB algorithm with DR up to *n*=39 jobs in a reasonable time to find the efficient solutions for the problem. In addition, to find good approximate solutions, we suggest two heuristic methods to solve the problem. The practical experiments prove the good performance of the two suggested methods.

Keywords: Single machine problem, total Completion time, maximum tardiness, Range of lateness, Branch and Bound.

Corresponding Author:
Manal Ghassan Ahmed
Mathematics Dept., College of Science, Mustansiriyah University, Baghdad, Iraq.
Email: manalghassan73@gmail.com

استخدام خوارزميات كفوءة لحل مسألة جدولة الماكنة ثلاثية المعايير

منال غسان احمد(1) **فائز حسن علي(2)**
(1،2) الجامعة المستنصرية، كلية العلوم، قسم الرياضيات، بغداد، العراق.

المستخلص

في هذا البحث سنتناول نموذج ماكنة منفردة متعددة المعايير. فليكن لدينا مسالة جدولة ماكنة (MSP) مكونة من *n* الاعمال لماكنة واحدة لتصغير دالة ثلاثية المعايير وهي: وقت الاتمام الكلي ($\sum C_j$)، مدى التاخير(R_L) و اعظم تاخير غير سلبي (T_{max}) وهي مسالة تعد من المسائل الصعبة (NP-hard) . اكثر من مسالة ثانوية تم اشتقاقها من مسالتنا الاصلية لغرض المناقشة والحل.

في الجزء النظري من هذا البحث، تم اثبات ان قاعدة (SPT) تعطي حل كفوء لمسالتنا وتم تطبيق بعض قواعد الهيمنة عليها. اما في الجانب العملي، تم تطبيق واحدة من اهم طرق الحصول على حل تام وهي خوارزمية التفرع والتقيد على لمسالتنا $(1//(\sum C_j, R_L, T_{max}))$حيث تم ايجاد حلول كفوءة للمسالة لـ *n*=18 وكذلك تم تنفيذ نفس الخوارزمية ولكن مع (DR) لـ *n*=39 لإيجاد حلول كفوؤة تامة وحلول كفوؤة تقريبية للمسالة في وقت مقبول.

الكلمات المفتاحية: ماكنة واحدة، تحسين متعدد الأغراض، إجمالي أوقات الإنجاز، أقصى تأخير.

ISSN (1681- 6870)

1. Introduction

Machine Scheduling problems (MSP) considered branch of the combinatorial optimization problems field, which it's defined as a decision making process that can be used on a many regular basis in various services industries and manufacturing. MSP deals with the allocation of resources to act over given time periods and its objective is to minimize one or many objectives [1]. Many fields that the scheduling theory has been of concerned; like computer science, manufacturing systems, transportation, industrial management, hospitals agriculture, and many other fields [2]. Tasks and resource are called jobs and machines respectively.

Scheduling is the process of assigning limited resources to a set of jobs over a period of time. The resources may be machines in a workshop, runways at an airport and crews at a construction site, as well as processing units in a computing environment and so on. The jobs may be operations in the production process, take-offs and landings at an airport, stages in a construction project, execution of computer programs and so on [3]. The goal of scheduling is to assign resources to the jobs such that one or more objectives are optimized. Within manufacturing scheduling, there are many different types of problem classes. These include single machine, parallel machine, flow shop and job shop [4].

The multicriteria scheduling problem has received significant attention in recent years and extensive survey of multicriteria are provided by Nagar et al. [5]. They show that two kinds of problems have been tackled. The first one deals with problems in which a lexicographical order of criteria is minimized. The studies by Smith [6], [5] and [7] are examples of hierarchical minimization problems. The second kind; simultaneous approach, there are two types, the first one typically generated all efficient schedules and select the one that yields the best composite objective function value of the two criteria. The second is to find the sum of these objectives. Several scheduling problems are considering the simultaneous minimization of various forms of objective functions. Hoogeveen (2005) [7] presents details survey of the most important results on multicriteria scheduling. The earliest study in simultaneous field has begun by Van Wassenhove and Gelders [8] they studied the efficiency with respect to the criteria the total completion times and the maximum tardiness in single machine problem. For more details about multicriteria (see [9,10,11]).

In this paper, we consider the problem of scheduling number of jobs (*n*) on a single machine to minimize a multicriteria objective function which be stated as follows: Each jobis to be processed on just one machine which can handle just one job at a time. For each job *j* there is a processing time and due date. All jobs are ready for processing at time zero. The aim is to find a set called Pareto optimal solutions set for the $1//F(\sum C_j, R_L, T_{max})$ problem.

In section two we will discuss the mathematical formulation of $1//(\sum C_j, R_L, T_{max})$problem. In section three the BAB will be proposed with new suggested upper and lower bound. Two heuristic methods are suggested to find near optimal solution for the suggested problem are introduced in section four. The practical and comparative results are introduced in section five. Lastly, in section six we will introduce the most important conclusions and some recommendations.

1.1 Important Notations

There are some notations are used in this paper:

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	:	Without DR.

1.2 Machine Scheduling Problem

In this paper we need some basic definitions.

Definition (1) [12]: Suppose we have set of all schedules S for a scheduling problem P, a schedule $\sigma \in S$, is said is called **feasible** if it satisfies all the constraints of the problem P.

Definition (2) [7]**:** A feasible schedule σ is called **Pareto optimal**, or **efficient** (non-dominated) with respect to the criteria f and g if there is absolutely no feasible schedule π such that both $f(\pi) \leq f(\sigma)$and $g(\pi) \leq g(\sigma)$, for at least one of the inequalities is strict.

Definition (3): (Shortest Processing Time(SPT) rule) [4]**:** 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 rule used to solve the problem $1//\sum C_j$.

Definition (4): Earliest Due Date (EDD) rule [7]: If the jobs are sequenced in non-decreasing order of due date (d_j)(i.e. $d_1 \leq d_2 \leq \cdots \leq d_n$). This rule is efficient to minimize the problem $1//T_{max}$.

Definition (5) [13]: The term "**optimize**" in a multicriteria resolution making problem indicates to a solution about which there is no way of improving or developing any objective without worsening the other objective.

1.3 Dominance Rule (DR)

Reducing the current sequence may be done by using several Dominance Rules (DR's). DR's usually specify some (all) parts of the path to obtain good value for objective function so they can be useful to determine whether a node in BAB method can be ignored before its lower bound (LB) is calculated. Clearly, DR's are particularly useful when a node can be ignored although it has a LB that is less than the optimum solution. The DR's are also useful within the BAB method to cut all nodes that are dominated by others. These improvements lead to very large decrease in the number of nodes to obtain the optimal solution**.**

Emmon's Theorem (1) [13]: For the 1/ /$\sum T_j$problem, if $p_i \leq p_j$ and $d_i \leq d_j$ then there exists an optimal sequencing in which job i sequencing before job j.

Definition (6) [13]: If G is a graph that has n vertices, then the **matrix** ***A*(*G*)=[*a*_ij_]**, whosei^{th}andj^{th}element is 1 if there is at least one edge between V_i and V_j and zero otherwise, is called the **adjacency matrix** of G, where:

$$a_{ij} = \begin{cases} 0, & \text{if } i = j \text{ or } j \nrightarrow i, \\ 1, & \text{if } i \rightarrow j, \\ a_{ij} \text{and } \bar{a}_{ij}, & i \leftrightarrow j. \end{cases}$$

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

Let $N = \{1,2, \dots, n\}$ be the set of jobs which are 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 three field classification suggested by Graham et al [3], the MSP denoted by $1//F(\sum C_j, R_L, T_{max})$. We will try to find the set of efficient solutions, for the machine which can be written for a given schedule $S = (1,2,\ldots,n)$ as:

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

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

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

In this section, we propose two techniques; classical Branch and Bound(BAB) or we can say BAB without DR (WDR) to determine a set of Pareto optimal solutions for *P*-problem. The BAB(WDR) steps are as follows:

Algorithm (1): BAB(WDR) Method

Step (1): **INPUT** n, $p_i$and d_j for $j = 1,2,\ldots,n$.

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

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

Step (4): At each node of the search tree of BAB method and for each partial sequence of jobs δ, compute a lower bound LB(δ) as follows: LB(δ) = cost of sequence jobs (δ) for the objective functions+cost of sequence jobs obtained by sequence the jobs in SPT rule.

Step (5): Branch from any node with LB ≤ UB.

Step (6): At the last level of search tree, we obtain a set of solutions, if $F(\delta)$ denote the outcome then δ is added to the set *S* unless it is dominated by the previously obtained efficient solutions in *S*, this process called Filtering *S*.

Step (7): **STOP**.

The BAB(WDR) can solved *P*-problem up to *n* =18 in a reasonable time.

Also in this section we introduce another BAB which depends on DR (BAB(DR)) to reduce the number of opened nodes which save time and increase the number of *n* for the solved problems. The main steps of this method are similar to BAB(WDR) with some different procedures. The BAB(DR) steps are as follows:

Algorithm (2): BAB(DR) Method

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

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

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

 ISSN (1681- 6870)

Step (4): At each node of the search tree of BAB method and for each partial sequence of jobs δ, compute a lower bound LB(δ) as follows: LB(δ) = cost of sequence jobs (δ) for the objective functions+cost of unsequence jobs obtained by sequence the jobs in SPT rule.
Step (5): Branch from each node with LB ≤ UB and $i \to j$.
Step (6): At the last level of search tree, we obtain a set of solutions, if $F(\delta)$denote the outcome then δ is added to the set *S* unless it is dominated by the previously obtained efficient solutions in *S*, this process called Filtering *S*.
Step (7): **STOP**.
The BAB(DR) we solve *P*-problem up to n=39 in a reasonable time.

4. Heuristic Method for *P*-problem

For the first heuristic method since the SPT rule solving the $1//\sum C_j$ problem, then calculate the objective function, and then put the second job in first place and the other jobs still arranged by SPT rule and calculate the objective function, and so on until obtain n sequences, the main steps of SPT-EDD-SCRLT are as follows:

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): Arrange jobs in SPT rule (σ_1), and calculate $F_{11}(\sigma_1)$; $\delta = \delta \cup \{F_{11}(\sigma_1)\}$.
Step (3):FOR *i=2,…,n*, put job *i* in the first position of σ_{i-1} to obtain σ_i and calculate $F_{1i}(\sigma_i)$;
$\delta = \delta \cup \{F_{1i}(\sigma_i)\}$.
END;
Step (4): Arrange jobsin EDD rule (π_1) , and calculate $F_{21}(\pi_1)$; $\delta = \delta \cup \{F_{21}(\pi_1)\}$.
Step (5): FOR *i=2,…,n*, put job *i* in the first position of $\pi_{i-1}$to obtain π_i and calculate $F_{2i}(\pi_i)$;
$\delta = \delta \cup \{F_{2i}(\pi_i)\}$.
END;
Step (6): Filter set δ to obtain as a set of efficient solution of *P*-problem
Step (7): OUTPUT The set of efficient solution δ.
Step (8): STOP.

The idea of the second heuristic method is summarized by finding a sequence sort with minimum p_j and d_j which is not contradiction with DR and calculate the objective function, The main steps of DR-SERLTareas follows:

Algorithm (4): DR_SCRLT Heuristic Method

Step (1): INPUT: n ,p_j and d_j, $j = 1,2,\dots,n$.
Step (2): Apply theorem (1) to find DR adjacency matrix A; $N = \{1,2,\dots,n\}, \delta = \emptyset$.
Step (3): Find a sequence σ_1 with minimum p_j which is not contradiction with DR(matrix A), if ∃ more than one job break tie arbitrary, $\delta = \delta \cup \{\sigma_1\}$.
Step (4): Find a sequence σ_2 with minimum d_j which is not contradiction with DR(matrix A), if ∃ more than one job break tie arbitrary, $\delta = \delta \cup \{\sigma_2\}$.
Step (5): Find the dominated sequence set $\delta'$from δ.
Step (6): Calculate $F(\delta)$.
Step (7): OUTPUT The set of efficient solution δ.
Step (8): END.

 ISSN (1681- 6870)

5. Practical Result of *P*-problems

The randomly values of p_j and d_j for all example are generated depending on the uniform distribution 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 results tables, we introduce some important abbreviations:

Ex : Example Number.
Av : Average.
NS : Number of efficient Solution.
ANS : Average number of efficient solution.
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 results of applying BAB(WDR) and BAB(DR) which are compared with CEM for *P*-problem, n=4:10 are shown in table (1).

Table(1): Comparison between BAB(WDR) and BAB(DR)with CEM for P-problem, *n*=4:10.

N	CEM			BAB(WDR)			BAB(DR)		
	OP	TIME	NES	MOF	TIME	NES	MOF	TIME	NES
	Av(F)	AT/S	ANES	Av(F)	AT/S	ANES	Av(F)	AT/S	ANES
4	(57.3,15.3,4.8)	R	3.8	(57.3,15.3,4.8)	R	3.8	(56.3,16.0,4.6)	R	2.8
5	(74.0,20.0,7.6)	R	10.2	(75.0,20.0,7.9)	R	9.2	(73.7,20.5,7.9)	R	8.0
6	(86.5,19.2,7.2)	R	10.4	(86.8,19.4,7.5)	R	9.0	(79.2,20.0,6.8)	R	7.6
7	(142.4,31.0,18.9)	R	16.4	(140.4,33.8,20.7)	R	9.8	(139.2,31.4,19.0)	R	10.6
8	(169.1,34.5,21.6)	1.4	12.4	(172.8,36.3,24.1)	R	9.6	(166.7,33.6,20.6)	R	7.6
9	(195.1,34.8,22.0)	11.6	15.4	(199.8,35.8,23.7)	R	11.2	(190.4,32.6,20.4)	R	9.6
10	(267.2,43.8,35.0)	139.9	5.0	(252.4,42.0,30.8)	R	8.4	(241.0,41.4,29.7)	R	7.2

From table (1), we notice that BAB(WDR) is more accurate to CEM results because its find all the solutions for P-problems with no matter that the optimal schedule which gives a solution is submit to the DR's or not.

In Table (2), a comparison has been made between BAB(WDR) and BAB(DR) for P-problem for n=11:18.

Table(2): Comparison between BAB(WDR) and BAB(DR) for P-problem, *n*=11:18.

n	BAB(WDR)			BAB(DR)		
	OP	TIME	NES	OP	TIME	NES
	Av(F)	AT/S	ANES	Av(F)	AT/S	ANES
11	(325.3,52.3,40.0)	R	10.4	(305.5,47.8,36.2)	R	9.6
12	(258.5,40.8,27.8)	R	11.0	(236.0,43.0,28.5)	R	8.2
13	(348.7,56.5,46.2)	R	11.4	(336.8,55.1,44.5)	R	10.4
14	(440.7,66.8,52.8)	R	12.2	(423.1,67.0,51.1)	R	12.0
15	(538.1,72.9,60.4)	R	12.6	(466.2,71.2,56.5)	R	10.4
16	(654.8,81.9,70.5)	R	10.6	(626.4,80.9,68.5)	R	8.6
17	(641.7,80.4,70.2)	R	11.2	(635.0,78.8,68.0)	R	7.6
18	(733.5,89.9,79.0)	7.5	14.4	(724.3,94.1,80.4)	R	8.2

For n=11:18, we notice that BAB(DR) starts to give minimum values for *P*-problem compared with results of BAB(WDR).

ISSN (1681- 6870)

The comparison results of SPT-EDD-SCRLT, DR-SCRLT with CEM, for *P*-problem, *n*=4:10 are shown in table (3).

Table (3): Comparison between SPT-EDD-SCRLT, DR-SCRLT with CEM for *P*-problem, n=4:10.

N	CEM			SPT-EDD-SCRLT			DR-SCRLT		
	OP	T	NS	MOF	T	NS	MOF	T	NS
	Av(F)	AT/S	ANS	Av(F)	AT/S	ANS	Av(F)	AT/S	ANS
4	(57.7,13.7,5.0)	R	3	(57.5,16.2,5.9)	R	3.2	(60.9,16.9,6.1)	R	2.4
5	(93.8,24.0,14.0)	R	5	(75.6,22.1,10.0)	R	4.8	(76.6,21.2,8.7)	R	3.0
6	(73.7,15.8,2.0)	R	10	(87.4,22.7,10.6)	R	5.0	(92.3,23.1,10.5)	R	2.6
7	(154.6,33.4,20.3)	R	14	(146.7,34.9,22.3)	R	6.2	(156.5,34.6,21.8)	R	3.2
8	(152.4,31.0,71.6)	1.4	10	(172.5,39.5,25.2)	R	6.4	(187.9,39.0,25.6)	R	3.4
9	(238.5,40.8,29.0)	13.1	12	(201.8,40.8,27.5)	R	6.8	(223.9,38.8,27.1)	R	3.2
10	(267.2,43.8,35.0)	139.9	5	(256.9,46.0,32.3)	R	5.2	(280.3,44.1,32.4)	R	3.0

Notice that the Heuristic SPT-EDD-SCRLT gives better results from DR-SCRLT compared with CEM for *P*-problem for *n*=4:10.

In table (4) we compare the results obtained from heuristic SPT-EDD-SCRLT and BAB(DR) for *P*-problem, n=11,15:(5):35,39.

Table (4): Results of comparison of BAB and SPT-EDD-SCSCRLT for P-problem, n=11,15:(5):35,39.

n	BAB(DR)			SPT-EDD-SCRLT		
	OP	T	NS	MOF	T	NS
	Av(F)	AT/S	ANS	Av(F)	AT/S	ANS
11	(305.5,47.8,36.2)	R	9.6	(332.8,59.3,44.8)	R	7.6
15	(466.2,71.2,56.5)	R	10.4	(558.9,78.0,64.4)	R	8.6
20	(896.5,101.4,88.6)	R	9.2	(938.3,106.3,90.7)	R	11.4
25	(1236.3,122.1,107.3)	R	11.8	(1312.6,125.6,108.9)	R	10.6
30	(1873.3,151.4,130.9)	R	7.6	(2064.9,154.5,135.9)	R	10.6
35	(2417.7,166.2,152.4)	2.1	14.2	(2559.5,177.3,155.2)	R	11.6
39	(3202.3,203.8,188.1)	4.6	13.0	(3337.1,211.5,189.1)	R	12.0

Table (5) introduces a comparison results between SPT-EDD-SCRLT and DR-SCRLT for *P*-problem for *n* =40, 70, 100, 400, 700, 1000.

Table (5): a comparison results between SPT-EDD-SCRLT and DR-SCRLT for P-problem for different n.

n	SPT-EDD-SCRLT			DR-SCRLT		
	MOF	NS	TIME	MOF	NS	TIME
	Av(F)	ANS	AT/S	Av(F)	ANS	AT/S
40	(3091.3,203.7,177.0)	13.4	R	(3588.4,187.5,173.3)	5.4	R
70	(10202.6,382.9,358.4)	13.4	R	(11439.6,362.0,349.5)	4.8	R
100	(20573.4,540.0,510.4)	14.2	R	(21979.4,514.4,499.7)	3.6	R
400	(327296.8,2193.3,2162.0)	14.0	1.9	(365685.4,2148.5,2143.1)	2.8	2.4
700	(1021767.1,3871.6,3839.5)	13.8	5.8	(1105158.4,3818.8,3814.4)	2.4	13.2
1000	(2040397.5,5499.1,5449.2)	13.4	13.4	(2136894.9,5421.4,5416.6)	2	36.0

Again notice that the Heuristic SPT-EDD-SCRLT gives better results from DR-SCRLT compared with CEM for *P*-problem for different *n*>39.

7. Conclusions and Future Works

1. In this paper, two techniques of BAB are proposed; with and without DR. BAB(WDR) is more accurate in $4 \leq n \leq 18$ and NS are larger than BAB(DR), that because its depend on condition $LB \leq UB$ only. But BAB(DR) is less accurate but its spend little CPU-time and BAB(DR) is more accurate in $n \leq 39$.
2. We suggest two good convenient heuristic methods for *P*-problem which are SPT-EDD-SCRLT and DR-SCRLT where they have good performance.
3. From *P*-problem we can derive more than one subproblems like $1//(\sum C_j + R_L + T_{max})$ and $1//Lex(\sum C_j, R_L, T_{max})$, and discussing their solving methods.
4. As future work, we suggesting to use local search methods (like particle swarm optimization, simulated annealing, Bees algorithm, genetic algorithm, …,etc) to find efficient and approximation solutions for *P*-problem for $n > 100$.

References

[1] Pinedo M. L., "Scheduling: theory, algorithms and systems," *New York: Springer*, 2008.
[2] Agin N., "Optimal seeking with 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, "A Comparison of Local Search Algorithm for Multicriteria Scheduling Problems,"*M.Sc. thesis, University of Al-Mustansiriyah, College of Science, Dept. of Mathematics*, 2016.
[10] Faez H. Ali and Shrmeen B. Abdul-Kareem, "Scheduling a Single Machine to Minimize Max Tardiness, Max Late Work and Total Late Work, "*Mathematics and Statistics Journal*, 3(1): 1-17, 2017.
[11] Hanan A. Chachan and Alaa S. Hameed, 2019, "Exact Methods for Solving Multi-Objective Problem on Single Machine Scheduling,"*Iraqi Journal of Science*, Vol. 60, No.8, pp: 1802-1813, DOI: 10.24996/ijs.2019.60.8.17, 2019.
[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, "Improving Exact and Local Search Algorithms for Solving Some Combinatorial Optimization Problems,"*Ph. D., Thesis, Mustansiriyah University, College of Science, Dept. of Mathematics*, 2015.