Al-Mustansiriyah Journal of Science
ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 33, Issue 2, 2022 DOI: http://doi.org/10.23851/mjs.v33i2.1091

Research Article Open Access

Exact Method with Dominance Rules for Solving Scheduling on a Single Machine Problem with Multiobjective Function

Manal Ghassan Ahmed*, Faez Hassan Ali

Department of Mathematics, College of Science, Mustansiriyah University, Baghdad, IRAQ

*Correspondent contact: manalghassan@uomustansiriyah.edu.iq

Article Info

Received
27/09/2021

Accepted
19/12/2021

Published
30/06/2022

ABSTRACT

The present article proposes an exact algorithm for the single-machine scheduling problem to minimize the sum of total completion times, range of lateness and maximum tardiness on a single machine $\left(1//\left(\sum C_{\sigma_j}+R_L+T_{max}\right)\right)$, where machine idle time is prohibited. In this paper, one of the multiobjective function problem for single criteria on just one machine is being studied. To obtain the optimal solution for the suggested problem, we propose to use Branch and Bound method (BAB) depending upon some dominance rules. This exact method used new technique to obtain three upper bounds (UB) and single lower bound (LB). The proposed BAB method proved its sufficiency by the practical results for n ≤ 15 in a reasonable time. Lastly, we proved a theorem as special case for our problem.

KEYWORDS: Multiobjective Problem (MOP), Branch and Bound (BAB) method, Upper Bound (UB), Lower bound (LB), Dominance Rules.

الخلاصة

في هذا المقال اقترحنا خوارزمية دقيقه لمسألة جدولة ماكنه واحده لتصغير مجموع وقت الاتمام الكلي ومدى التأخير واكبر تأخير على ماكنة واحدة بحيث وقت توقف الماكنه غير مسموح به. في هذا البحث درسنا واحدة من مسائل دالة متعددة الاهداف لمعيار واحد على ماكنة واحدة. لايجاد الحل الامثل للمسألة المقترحة استخدمنا طريقة التفرع والتقيد بالاعتماد على قواعد الهيمنة باسلوب جديد لايجاد القيود العليا والقيد الادنى وقد اثبت كفاءة هذه الطريقة بالنتائج العمليه الى حد عدد الاعمال يساوي 15 عمل في وقت معقول. واخيرا برهنا نظرية كحالة خاصة لمسألتنا.

INTRODUCTION

The **Machine Scheduling Problems (MSP)** plays a very important role in most manufacturing and production systems as well as in most information processing environment. The scheduling theory plays a great role in solving the machine scheduling problem (MSP) which are work in many fields for instance production facilities. The basic concept of MSP is interpreted for each of objectives, which we called it jobs, an interval of execution on a single machine, where all constraints are satisfied. The solution of MSP is called schedule can be considered a best possible to minimize the multiobjective function [1]. To improve the society in manufacturing, we use the production process to end the goods manufacturing for parts of them or some components and raw materials. The following are called the customers' expectations:

1. The quality of the product,
2. The safety environmental,
3. The attractive of product,
4. The truthful products…, etc.

So, the decision maker is must study those expectation well to minimize the cost, so they always must monitor the performance, the objectives and the priority levels. There are many objectives must be satisfied like number of late jobs, total lateness and completion time. The single machine, closed shop, flow shop, open shop and hybrid job shop are as considered as scheduling [2]. In this study, we introduced one of single machine problems, so that $\left(1//\left(\sum C_{\sigma_j}+R_L+T_{max}\right)\right)$ is minimized. More specifically, we consider that the set of jobs $N=\{1,2,\dots,n\}$ are considered on a single machine, each job $j\in$ N has positive integer p_j and d_j (where p_j is the processing time of job j and d_j is the due date of job j). Machine idle time is not permitted and the machine cannot process more than one job at a time.

There are many papers focus on multi-objective function single MSP, Ali and Abdul-Kareem (2017), try to solve multicriteria objective function for single machine to minimize $1//(T_{max}, V_{max}, \sum V_i)$, they suggested heuristic and exact method (BAB) to solve their problem [3]. Abdul–Razaq and Motair (2018) they consider single MSP to minimize the sum of four cost functions; total completion times, total tardiness, maximum tardiness, and maximum earliness. The minimization based on two types, in the first one they study some special cases including lexigraphical minimization of problem. In the second type they minimize the four cost functions simultaneously and propose algorithm to find the set of efficient solution for the discussed problem [4]. Chachan and Jaafar (2019) present BAB method to minimize the sum of total completion time, total tardiness, total earliness, number of tardy jobs and total late work with unequal release dates. they proposed six heuristic methods for account upper bound (UB). Also, to obtain lower bound (LB) to this problem they use Moore and Lawler's algorithm. And some dominance rules were suggested, with two special cases [5].

Abbas (2019), Study a multi-objectives single MSP, the objective is to minimize four cost functions ($\sum C_i+\sum U_i+\sum T_i+T_{max}$) by BAB method and local search methods (LSMs) and developed a simple heuristic method to solve the considered problem [6]. Jawad, Ali and Hasanain (2020), investigated in their paper some methods to solve one of the multi-criteria machine scheduling problems. our discussed problem is the total completion time and the total earliness jobs, they solved our problem by proposed some heuristic methods which provided good results. They applied (BAB) method with new suggested upper and lower bounds to solve the discussed problem, which produced exact results for $n \leq 20$ in a reasonable time [7].

The rest of this paper is organized as follows: in section 2, the multiobjective problem definition is described, in section 3, Dominance rule for adjacent jobs is presented. in section 4, Dominance Rules are presented. In section 5, we describe the decomposition of our problem (P), in section 6, we introduced the BAB method with DR that we used in this paper to find the optimal solution, Comparisons Results for P-Problem are introduced in section 7**.** Lastly, Conclusion and Future Work are presented in section 8.

We define the most important objective function use in our study:

$C_j = \sum_{k=1}^{j} p_k$

$L_j = C_j - d_j$.

$L_{min} = \min_j \{L_j\}$.

$L_{max} = \max_j \{L_j\}$.

$R_L = L_{max} - L_{min}$.

$T_j = \max_j \{L_j, 0\}$ and $T_{max} = \max_j \{T_j, 0\}$.

In this paper, we will take in consideration some important rules like: Shortest Processing Time (SPT Rule [8]), Earliest Due Date (EDD rule [9]), Minimum Slack Time (MST rule [10])

Lemma (1) [11]: For the $1//(\sum C_j, \sum T_j)$ 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*.

Formulation of the Discussion Problem Mathematically

The single MSP under consideration can be defined as follows: for a given schedule $\sigma = (1,2, \dots, n)$:

$$\left.\begin{array}{l}
V = Min\, \{\sum C_j + R_{L(\sigma)} + T_{max}\} \\
\text{s.t.} \\
C_1 \geq p_{\sigma(1)}, \\
C_j = C_{(j-1)} + p_{\sigma(j)}, j = 2,3, \dots, n \\
L_j = C_j - d_{\sigma(j)}, j = 1,2, \dots, n \\
T_j \geq C_j - d_{\sigma(j)}, j = 1,2, \dots, n. \\
R_L(\sigma) = L_{max}(\sigma) - L_{min}(\sigma), \\
C_j, T_j \geq 0, j = 1,2, \dots, n.
\end{array}\right\} \quad P$$

The *target of P- Problem is* to find the best arrangement of the jobs on a single machine to minimize $\left(\sum C_{\sigma_j} + R_L + T_{max}\right)$, $\sigma \in S$ (where S is the set of all feasible solutions).

Dominance Rules for Adjacent Jobs

Let $F(S)$ and $R(S)$ represent the total completion times and range of lateness for a particular schedule S, Then:

$F(S) = \sum_{j=1}^{n} C_{(S(j))}$,

$R_L(S) = max\{ L_j(S)\} - min\,\{L_j(S)\}$,

ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 33, Issue 2, 2022 DOI: http://doi.org/10.23851/mjs.v33i2.1091

$T_{max}(S) = max\{0, L_j\}(S)$

Hence the problem becomes one of determining a schedule S which minimizes the following objective function:

$$Z(s) = F(s) + R_L(S) + T_{max}(S) \quad (1)$$

Let S be a schedule in which job i appears before job j. Let S_1 be a schedule which is obtained from S by interchanging i and j only. Let T_B be the sum of the process times of all the jobs scheduled before jobs i and j and T_B is the same in both schedules S and S_1. Let $L_i(S)$ and $L_j(S)$ be the lateness of job i and j in a schedule S and a similar definition apply to $L_i(S_1)$ and $L_j(S_1)$ for S_1.

Let $F(S)$ and $F(S_1)$ be the sum of total completion times of all jobs in schedules S and S_1 respectively. Let F_0 denote the sum of completion times of all jobs $J-(i,j)$, then:

$$F(S) = F_0 + (T_B + p_i) + (T_B + p_i + p_j)$$

$$F(S_1) = F_0 + (T_B + p_j) + (T_B + p_j + p_i)$$

Hence by subtraction the above two relations

$$F(S) - F(S_1) = p_i - p_j.$$

Let $R(S)$ and $R(S_1)$ be the range of lateness measures of schedules S and S_1 respectively. If

$L = max\{L_k \mid k \in J - (i,j)\}$ and

$L_0 = min\{L_k \mid k \in J - (i,j)\}$.

Let $T(S)$ and $T(S_1)$ represent the maximum tardiness measures in schedules S and S_1 respectively.

If $T_{max}(S) = max\{T_k \mid k \in J - (i,j)\}$ and $T_{max}(S_1) = max\{L_k \mid k \in J - (i,j)\}$ in either schedule.

Let $max\{L, L_i(S), L_j(S)\} = M$

$min\{L_0, L_i(S), L_j(S)\} = N$

$max\{L, L_j(S_1), L_i(S_1)\} = W$

$min\{L_0, L_j(S_1), L_i(S_1)\} = Q$

Then:

$$R_L(S) = max\{L, L_i(S), L_j(S)\} - min\{L_0, L_i(S), L_j(S)\} = M - N$$

$$R_L(S_1) = max\{L, L_j(S_1), L_i(S_1)\} - min\{L_0, L_j(S_1), L_i(S_1)\} = W - Q$$

Now under $S: L_i = T_B + p_i - d_i,\ L_j = T_B + p_i + p_j - d_j$

Under $S_1: L_{1j} = T_B + p_j - d_j,\ L_{1i} = T_B + p_j + p_i - d_i$

Then the objective function under the schedule S and S_1 will be given by:

$$Z(S) = F(S) + R(S) + T_{max}(S) = F(S) + (M - N) + T_{max}(S)$$

$$Z(S_1) = F(S_1) + R(S_1) + T_{max}(S_1) = F(S_1) + (W - Q) + T_{max}(S_1)$$

Then:

$$Z(S) - Z(S_1) = (p_i - p_j) + (M - W) - (N - Q) + T_{max}(S) - T_{max}(S_1) \quad (2)$$

Now by definition:

$$L_i(S) = T_B + p_i - d_i \quad (3)$$

$$L_j(S) = T_B + p_i + p_j - d_j \quad (4)$$

$$L_j(S_1) = T_B + p_j - d_j \quad (5)$$

$$L_i(S_1) = T_B + p_j + p_i - d_i \quad (6)$$

Theorem (1): Given $p_i > p_j$ then we have the following cases:

1. $Z(S) - Z(S_1) \le p_i - p_j$ if $d_i \le d_j$.
2. $Z(S) - Z(S_1) \le p_i - p_j + (d_i - d_j)$if $d_i > d_j$.

Proof:

From relations (3-6), we have:

$M = \max\{L, L_i(S), L_j(S)\}$

$N = min\{L', L_i(S), L_j(S)\}$

$W = max\{L, L_j(S_1), L_i(S_1)\}$

$Q = min\{L', L_j(S_1), L_i(S_1)\}$

Then:

$$Z(S) - Z(S_1) = p_i - p_j + (M - W) - (N - Q) + T_{max}(S) - T_{max}(S_1)$$

$T_{max}(S) = T_{max}(S_1)$

So we have the following cases:

Case (1): $d_i \le d_j$

From relation (3-6):

$$L_j(S_1) \le L_i(S) \le L_i(S_1)$$

$$L_j(S_1) \le L_j(S) \le L_i(S_1)$$

$min\{L_j(S_1), L_i(S_1)\} \ge min\{L_i(S), L_j(S)\}$

$i.e.,\ Q \ge N$

$L_j(S) = T_B + p_i + p_j - d_j \le L_i(S_1) = T_B + p_j + p_i - d_i$

$L_i(S) \leq L_i(S_1)$
So $max\{L_j(S), L_i(S)\} \leq max\{L_i(S_1), L_j(S_1)\}$
$i.e., \quad M \leq W$.
So from relation (2) we obtain $Z(S) - Z(S_1) \leq p_i - p_j$
$Z(S) - Z(S_1) = p_i - p_j + (\text{M}-\text{W}) - (\text{N}-\text{Q}) + T_{max}(S) - T_{max}(S_1)$
$Z(S) - Z(S_1) \leq (p_i - p_j)$.

Case (2): $\boldsymbol{d_i > d_j}$
$min\{L_j(S_1), L_i(S_1)\} \leq min\{L_i(S), L_j(S)\}$
$\quad i.e., \quad Q \leq N$
$M - W = max\{L, L_i(S), L_j(S)\} - max\{L, L_j(S_1), L_i(S_1)\}$
$L_j(S_1) \leq L_i(S_1)$
$L_j(S) > L_i(S)$
$M - W = max\{L, L_j(S)\} - max\{L, L_i(S_1)\}$
$\quad L_j(S) > L_i(S_1)$

1. $L \leq L_i(S_1) \leq L_j(S)$, then $M - W = L_j(S) - L_i(S_1) = T_B + p_i + p_j - d_j - (T_B + p_j + p_i - d_i) = d_i - d_j$
2. $L_i(S_1) < \text{L} \leq L_j(S) = M - W = L_j(S) - L \leq L_j(S) - L_i(S_1)$.
3. $L_i(S_1) \leq L_j(S) < L = M - W = L - L = 0$.

Therefore $M - W$ is not greater than $d_i - d_j$,
Hence (2) $Z(S) - Z(S_1) \leq (p_i - p_j) + (d_i - d_j)$.
The following example explains case (1) of theorem (1).

Example (1):

	1	**2**	**3**	**4**	**5**
$\boldsymbol{p_j}$	2	3	**5**	**4**	6
$\boldsymbol{d_j}$	3	5	**6**	**8**	7

Let's choose $S = (1, 2, 3, 4, 5)$, let $J_i = J_3$ and $J_j = J_4$, $S_1 = (1, 2, 4, 3, 5)$.
For sequence S we have:

$\boldsymbol{S}$ | $\boldsymbol{j}$	**1**	**2**	**3**	**4**	**5**
$\boldsymbol{p_j}$	2	3	**5**	**4**	6
$\boldsymbol{d_j}$	3	5	**6**	**8**	7
$\boldsymbol{C_j}$	2	5	**10**	**14**	20
$\boldsymbol{L_j}$	-1	0	**4**	**6**	13
$\boldsymbol{T_j}$	0	0	**4**	**6**	13

For sequence S_1 we have:

$\boldsymbol{S_1}$ | $\boldsymbol{j}$	**1**	**2**	**4**	**3**	**5**
$\boldsymbol{p_j}$	2	3	**4**	**5**	5
$\boldsymbol{d_j}$	3	5	**8**	**6**	7
$\boldsymbol{C_j}$	2	5	**9**	**14**	20
$\boldsymbol{L_j}$	-1	0	**1**	**8**	13
$\boldsymbol{T_j}$	0	0	**1**	**8**	13

$L_j(S_1) = 1 \leq L_i(S_1) = 8$
$L_j(S) = 6 > L_i(S) = 4$
$L_j(S) = 6 \leq L_i(S_1) = 8$
$max\{L_i(S), L_j(S)\} \leq max\{L_j(S_1), L_i(S_1)\}$
$\rightarrow max\{4, 6\} \leq max\{1, 8\}$
$min\{L_j(S_1), L_i(S_1)\} \leq min\{L_i(S), L_j(S)\}$
$min\{1, 8\} \leq min\{4, 6\}$
$Z(S) = F(S) + R_L(S) + T_{max}(S)$ and $Z(S_1) = F(S_1) + R_L(S_1) + T_{max}(S_1)$.
Then $Z(S) = 51 + 14 + 13 = 78$ and $Z(S_1) = 50 + 14 + 13 = 77$.
$Z(S) - Z(S_1) = 78 - 77 = 1 \leq p_i - p_j = 5 - 4 = 1$.
This means:
$Z(S) \leq Z(S_1)$ if $p_i > p_j$ and $d_i \leq d_j$.

The following example explains case (2) of theorem (1).

Example (1):

	1	2	3	4	5
p_j	2	3	**5**	**4**	6
d_j	3	5	**8**	**6**	7

Let choose $S = (1, 2, 3, 4, 5)$, let $J_i = J_3$ and $J_j = J_4$, $S_1 = (1, 2, 4, 3, 5)$.
For sequence S we have:

$\boldsymbol{S}$ | $\boldsymbol{j}$	**1**	**2**	**3**	**4**	**5**
$\boldsymbol{p_j}$	2	3	**5**	**4**	6
$\boldsymbol{d_j}$	3	5	**8**	**6**	7
$\boldsymbol{C_j}$	2	5	**10**	**14**	20
$\boldsymbol{L_j}$	-1	0	**2**	**8**	13
$\boldsymbol{T_j}$	0	0	**2**	**8**	13

For sequence S_1 we have:

$\boldsymbol{S_1}$ | $\boldsymbol{j}$	**1**	**2**	4	3	**5**
$\boldsymbol{p_j}$	2	3	4	5	5
$\boldsymbol{d_j}$	3	5	6	8	7
$\boldsymbol{C_j}$	2	5	9	14	20
$\boldsymbol{L_j}$	-1	0	3	6	13
$\boldsymbol{T_j}$	0	0	3	6	13

$Z(S) = F(S) + R_L(S) + T_{max}(S)$ and $Z(S_1) = F(S_1) + R_L(S_1) + T_{max}(S_1)$.
Then $Z(S) = 51 + 14 + 13 = 78$ and
$Z(S_1) = 50 + 14 + 13 = 77$.
$p_i - p_j = 5 - 4 = 1$.

$d_i - d_j = 8 - 6 = 2.$
Then $p_i - p_j - (d_i - d_j) = 3$
$Z(S) - Z(S_1) = 78 - 77 = 1 \leq 1 + 2 = 3.$
This means: $Z(S) \leq Z(S_1)$
$p_i > p_j$ and $d_i > d_j$.

Dominance Rules

Dominance Rules (DR's) play a good role to obtain a good and fast approach to reducing the current sequence to be used specially in BAB method (for more details see [14]).

Decomposition of $P - Problem$

The $P - Problem$ can be decomposed into three subproblems (SP_1), (SP_2) and (SP_3), let these subproblems be as follows:

$$SP_1 \left\{ \begin{array}{l} V_1 = Min\,\{\sum C_j\} \\ \text{s.t.} \\ C_1 \geq p_{\sigma(1)}, \\ C_j = C_{(j-1)} + p_{\sigma(j)} \quad, j = 2,3,\dots,n \\ C_j \geq 0, \quad j = 1,2,\dots,n. \end{array} \right.$$

$$SP_2 \left\{ \begin{array}{l} V_2 = Min\,\{R_L\} \\ \quad = min\{L_{max} - L_{min}\} \\ \text{s.t.} \\ C_1 \geq p_{\pi(1)}, \\ C_j = C_{(j-1)} + p_{\pi(j)}, \quad j = 2,3,\dots,n. \\ L_j = C_j - d_{\pi(j)}, \quad j = 1,2,\dots,n. \\ L_{max} \geq L_{min}, \\ C_j \geq 0, \quad j = 1,2,\dots,n. \end{array} \right.$$

$$SP_3 \left\{ \begin{array}{l} V_3 = Min\,\{T_{max}\} \\ \quad = min\left\{max\{T_j\}\right\} \\ \text{s.t.} \\ C_1 \geq p_{\delta(1)}, \\ C_j = C_{(j-1)} + p_{\delta(j)}, j = 2,3,\dots,n. \\ T_j \geq C_j - d_{\delta(j)}, \quad j = 1,2,\dots,n. \\ T_j \geq 0, \quad j = 1,2,\dots,n. \end{array} \right.$$

Solving $P - Problem$ using BAB

BAB is one of the most important tools in the construction the optimal solution for discrete NP-hard optimization problems. A BAB algorithm searches the complete space of solutions for a given problem for the optimal solution. But, Because of the exponentially increasing number of possible solutions, explicit enumeration is normally impossible. So, the use of bounds for the function to be minimized (maximized) with the value of the current best solution helps the algorithm work on parts of the solution space [12]. BAB method is an exact method that widely used in MSPs to obtain the optimal solution. In this study, we used BAB to find the optimal solution for P-Problem.

BAB with Decomposition Technique

In order to describe this technique, we have to introduce the following theorem for decomposition procedure.

Theorem (2): If V_1, V_2, V_3 and V are the minimum objective function values of the subproblems (SP_1), (SP_2), (SP_3) and P-Problem respectively. Then $V_1 + V_2 + V_3 \leq V$.

Proof

Let σ be an optimal schedule to (P) and
$V = S_1 + S_2 + S_3$
Where
$S_1 = \sum_{j\in N} C_{\sigma(j)}$, $S_2 = T_{max}(\sigma)$ and
$S_3 = R_L(\sigma) = Tmax(\sigma)$.
Clearly σ is feasible schedule to subproblems (SP_1), (SP_2) and (SP_3).
Hence $S_1 \geq V_1, S_2 \geq V_2$ and $S_3 \geq V_3$.
This yields that
$V = S_1 + S_2 + S_3 \geq V_1 + V_2 + V_3$.
From theorem (2) we can derive a new lower bound LB for problem (P) to apply new BAB technique.

Derivation of Upper Bound

We can find upper bound (UB) for problem (P) by using:

1. UB_1 depends on σ = SPT rule, then:

$$UB_1 = \sum_{j=1}^{n} C_{\sigma(j)} + R_L(\sigma) + T_{max}(\sigma) \tag{7}$$

2. UB_2 depends on π = MST rule, then:

$$UB_2 = \sum_{j=1}^{n} C_{\pi(j)} + R_L(\pi) + T_{max}(\pi) \tag{8}$$

3. UB_3 depends on δ = EDD rule, then:

$$UB_3 = \sum_{j=1}^{n} C_{\delta(j)} + R_L(\delta) + T_{max}(\delta) \tag{9}$$

Then:

$$UB = min\,\{UB_1, UB_2, UB_3\} \tag{10}$$

Derivation of Lower Bound

A lower bound (LB) for problem (P) is based on the decomposition of this problem which is mentioned in section 4. Now we calculate V_1 to be the LB for subproblem (SP1) problem, V_2 to be the LB for subproblem (SP2) and V_3 to be the LB for subproblem (SP3) then applying theorem (2), then we obtain a LB for problem (P).

For subproblem (SP1), we obtained the LB by sorting the jobs by $\sigma =$ SPT rule and calculate:

$$LB(SP_1) = \sum_{j=1}^{n} C_{\sigma(j)} \tag{11}$$

For subproblem (SP2), we obtained the LB by sorting the jobs by $\pi =$ MST rule and calculate:

$$LB(SP_2) = R_L(\pi) \tag{12}$$

For subproblem (SP3), we obtained the LB by sorting the jobs by $\boldsymbol{\delta} =$ EDD rule and calculate:

$$LB(SP_3) = T_{max}(\delta) \tag{13}$$

Then:

$$LB = LB(SP_1) + LB(SP_2) + LB(SP_3) \tag{14}$$

Where the LB is the LB for unsequence jobs.

The suggested new BAB method depends on two techniques; the first is represented by using Lemma(1) to find the DRs for the problem. While the second one is the decomposition technique which is introduced by theorem (2). The new BAB is called BAB depends on DR and decomposition techniques (BABDRDT) is a suggested method to obtain optimal solution for $P - Problem$. The BABDRDT algorithm is as follows:

BABDRDT Algorithm

Step(0):INPUT:n, p_j and $d_j, j = 1, \ldots, n$**,**

Calculate the matrix $A(G)$**,** $lev = 0$**;**

Step(1): Calculate UB at the parent node of the search tree: by using relation (7),(8) and (9). **Then the UB can be calculated as in relation (10).**

Step(2): l$ev = lev + 1$**;**

For each node of the search tree of BABDRDT i.e. for each partial sequence of jobs (say σ), compute $LB(\sigma)$ where σ is the partial sequence in every node of the tree as follows: $LB(\sigma)$=cost of sequence jobs (σ) for the objective functions + cost of unsequence jobs obtained by **relation (14).**

Step(3): If LB ≤ UB then branch from it which must be subject to $A(G)$.

Step(4): if $lev \leq n - 1$ goto Step(2).

Step(5): At the last level of the tree, we get an optimal solution for *P*- problem.

Step(6): Stop.

Comparisons Results for $\boldsymbol{P - Problem}$

For each number of jobs, we generated (5) examples, with the $p_j \in \{1,2, \ldots, 10\}$ and $d_j \in \{1,2, \ldots, 70\}$ which are generated uniformly under condition $d_j \geq p_j$ for $j = 1, \ldots, n$. To understand the comparisons tables, we introduce the following notations which are used in the tables of results:

n : Number of jobs.

Av : Average values of (5) examples.

Gav: General Average of Av.

MT/s : Mean of CPU-Time for (5) examples per second.

OV : Optimal Value of *P*- problem for (5) examples.

BS : best solution Value of *P*- problem for (5) examples.

T : T ∈ [0,1), where T is real number .

F : Objective Function of *P*- problem.

Table 2. comparison the results between CEM and BABDRDT for $n = 4:11$.

n	CEM		BABDRDT	
	OV	**TIME**	**OV**	**TIME**
	Av(F)	***MT/s***	***Av(F)***	***MT/s***
4	74.6	*T*	74.6	*T*
5	94.6	*T*	94.6	*T*
6	106.6	*T*	106.6	*T*
7	186.0	*T*	186.0	*T*
8	218.0	*T*	218.0	*T*
9	245.0	*T*	245.0	*T*
10	313.6	6.9	313.6	*T*
11	403.0	80.3	403.0	*T*
$\boldsymbol{Gav}$	**205.2**	**10.9**	**205.2**	***T***

From Table 2, we notice that the CPU-Time between CEM and BABDRDT are the identical but for $n = 10$ and 11, CEM is taken long time than BABDRDT.

ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 33, Issue 2, 2022 DOI: http://doi.org/10.23851/mjs.v33i2.1091

Table 3. Comparison the results between BABDRDT and TTHM [13] for $n = 4:15$.

N	BABDRDT		TTHM	
	OV	TIME	BS	TIME
	Av(F)	*MT/s*	*Av(F)*	*MT/s*
4	74.6	*T*	76.4	*T*
5	94.6	*T*	100.4	*T*
6	106.6	*T*	113.8	*T*
7	186.0	*T*	194.0	*T*
8	218.0	*T*	228.6	*T*
9	245.0	*T*	253.6	*T*
10	313.6	*T*	317.0	*T*
11	403.0	*T*	412.2	*T*
12	299.6	1.6	307.0	*T*
13	424.4	5.7	434.2	*T*
14	557.4	8.6	560.8	*T*
15	639.0	329.5	653.6	*T*
Gav	**205.2**	**28.8**	**304.3**	***T***

From Table 3, we notice that the results of applying BAB(SR) are better than the results of TTHM.

Table 4. Comparison the results between BABDRDT, PSO [14] and BA [14] for $n = 4:15$.

n	BABDRDT		PSO		BA	
	OV	TIME	BS	TIME	BS	TIME
	Av(F)	*MT/s*	*Av(F)*	*MT/s*	*Av(F)*	*MT/s*
4	74.6	*T*	74.6	*T*	74.6	*T*
5	94.6	*T*	94.6	*T*	94.6	*T*
6	106.6	*T*	106.6	1.1	106.6	*T*
7	186.0	*T*	186.6	1.2	186.6	*T*
8	218.0	*T*	218.0	1.2	218.0	*T*
9	245.0	*T*	245.0	1.3	245.0	*T*
10	313.6	*T*	313.6	1.5	313.8	*T*
11	403.0	*T*	403.4	1.6	403.3	*T*
12	299.6	1.6	299.8	1.7	303.2	*T*
13	424.4	5.7	425.0	1.7	430.0	*T*
14	557.4	8.6	557.4	1.9	569.2	*T*
15	639.0	329.5	639.8	2.0	662.0	*T*
Gav	**205.2**	**28.8**	**296.7**	**1.3**	**300.6**	***T***

We notice from Table 4 the results of applying BABDRDT are the best among the results of applying PSO and BA, and the results of PSO are closed to BABDRDT.

CONCLUSION AND FUTURE WORK

In the present study, one of the multiobjectives function a single machine (MSP) is considered with dominance rules i.e., $\left(1//\left(\sum C_{\sigma_j} + R_L + T_{max}\right)\right)$, we used BAB algorithm with DR to find the optimal solution up to $n = 15$ jobs. The results of applying BAB algorithm are comparison with CEM, TTHM, PSO and BA. We proved important theorem as a special case of the P- problem.

We will suggest some problems to be discussed and analyzed as future work:

1. $1//Lex\left(\sum C_j + R_L + T_{max}\right)$.
2. $1//Lex\left(R_L + \sum C_j + T_{max}\right)$.
3. $1//Lex\left(T_{max} + \sum C_j + R_L\right)$.

REFERENCES

[1] Hoogeveen H., "Multicriteria Scheduling, Department of Computer, Utrecht University, P.O. Box 80089, Utrecht 3508TB, (2005). Netherlands.

[2] Poongothai V., Godhandaraman P.and Arockia Jenifer A.,"Single Machine Problem for Minimizing Total Tardiness of a Weighted Jobs in a Batch Delivery System, Stochastic Rework and Reprocessing Times. 020132(2019). AIP Conference Proceeding, 2112.

https://doi.org/10.1063/1.5112317

[3] Faez H. Ali and Shrmeen B. Abdul-Kareem, "Scheduling a Single Machine to Minimize Max Tardiness,Max Late Work and Total Late Work",2017,Mathematics and Statistics Journal, 3(1): 1-17.

[4] Abdul-Razaq, T. S., and Motair, H. M., "Solving Four Cost Multi-Objective Scheduling Problem Simultaneously", (2018), Journal of Kerbala University, Vol. 16(1), pp. 77-88.

[5] Hanan Ali Chachan and Hussein Abdullah Jaafar, "Exact Solutions for Minimizing cost Function with Five Criteria and Release Dates on Single Machine". 2020, Ibn Al-Haitham Journal For Pure and Appl. Sci.33(3), pp.140-157. Doi: 10.30526/33.3.2479

[6] D. A. Abbass, "Using Branch and Bound and Local Search Methods to solve Multi-Objective Machine Scheduling Problem". First International Conference of Computer and Applied Sciences (CAS), (2019), pp:63-66.

https://doi.org/10.1109/CAS47993.2019.9075460

[7] A. A. Jawad, F. H. Ali and W. S. Hasanain "Using Heuristic and Branch and Bound Methods to Solve a Multi-Criteria Machine Scheduling Problem", Iraqi Journal of Science, 2020, Vol. 61, No. 8, pp: 2055-2069,

https://doi.org/10.24996/ijs.2020.61.8.21

[8] Smith W. E.,"Various optimizers for single– stage production", Navel Research Logistics Quarterly, 1956, 3, 59-66.

https://doi.org/10.1002/nav.3800030106

[9] Hoogeveen, H., "Invited Review Multicriteria Scheduling", European Journal of Operation Research, , 2005, 167:592-623.

https://doi.org/10.1016/j.ejor.2004.07.011

[10] Hoogeveen J.A., "Minimizing maximum earliness and maximum lateness on a single machine", In Proceedings of the 1st Integer Programming and Combinatorial Optimization Conference (pp. 283-295).

[11] F. Ali and T. Abdul-Razaq, "Combinatorial Optimization Problems: Modeling and Solving", Lap Lambert Academic Publishing, 2016, ISBN-13: 978-3-659-93685-2.

[12] Hafed M. "Solving Single and Flow Shop Scheduling Multicriteria Problems Using Branch and Bound and Local Search Algorithms", PhD Thesis Dept. of Mathematics, College of Science, Mustansiriyah University, (2018).

[13] Faez H. Ali and Manal G. Ahmed, "Optimal and Near Optimal Solutions for Multi Objective Function on a Single Machine" International Conference on computer Science and Software Engineering (CSASE), IEEE, Duhok Iraq, 2020, pp. 152-156.

DOI:10.1109/CSASE48920.2020.9142053

[14] Faez H. Ali and Manal G. Ahmed, "Local Search Methods for Solving Total Completion Times, Range of Lateness and Maximum Tardiness Problem",6th international engineering conference", Sustainable Technology and Development", (IEC-2020), IEEE, Erbil Iraq, 2020, pp. 103-108.

DOI: 10.1109/IEC49899.2020.9122821

How to Cite

M. G. Ahmed and F. H. . Ali, "Exact Method with Dominance Rules for Solving Scheduling on a Single Machine Problem with Multiobjective Function", *Al-Mustansiriyah Journal of Science*, vol. 33, no. 2, pp. 56–63, Jun. 2022.