Al-Mustansiriyah Journal of Science
ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 33, Issue 3, 2022 DOI: http://doi.org/10.23851/mjs.v33i3.1135

Research Article Open Access

Solving Bi-Criteria and Bi-Objectives of Total Tardiness Jobs Times and Range of Lateness Problems Using New Techniques

Faez Hassan Ali[1*], Riyam Noori Jawad[2], Wadhah Abdulleh Hussein[3]

[1]Mathematics department, College of Science, Mustansiriyah University, IRAQ.
[2]Total Quality Management Techniques department, Technical College of Management, Middle Technical University, IRAQ.
[3]Mathematics department, College of Science, University of Diyala, IRAQ.

*Correspondent contact: faezhassan@uomustansiriyah.edu.iq

Article Info

Received
14/02/2022

Accepted
28/03/2022

Published
25/09/2022

ABSTRACT

In this paper, we proposed new techniques to solve one of the important fields of the Combinatorial Optimization Problem, which is the problem of machine scheduling. The problem which wants to be solved in this paper is the total tardiness times and range of lateness. For bi-criteria we have $(1//(\sum T_j, R_L))$ while for bi-objective we have $(1//(\sum T_j + R_L))$. In order to solve the suggested two problems, some new exact and approximation methods are suggested which are produced good results. The results which are obtained from applying the newly proposed methods are compared with the exact method; like Complete Enumeration Method (CEM), then compared results of the heuristics with each other's to obtain the most efficient method.

KEYWORDS: Bi-criteria and multi-objective machine scheduling problems, branch and bound, tardiness time, lateness, range of lateness.

الخلاصة

في هذا البحث تم اقتراح تقنيات جديدة لحل واحدة من اهم حقول المسائل الامثلية التوافقية وهي مسالة جدولة الماكنة ثنائية المعايير وثنائية الاهداف. ان المسألة المراد حلها في هذا البحث تتمثل بمسألة مجموع اوقات التاخيرالاسلبي ومدى التاخير. فيما يخص مسالة ثنائية المعايير سيتم مناقشة المسالة $(1//(\sum T_j, R_L))$ اما فيما يخص مسالة ثنائية الاهداف سيتم مناقشة المسالة $(1//(\sum T_j + R_L))$. ولحل هاتين المسالتين، تم اقتراح بعض الطرق المضبوطة والتقريبية والتي اعطت نتائج جيدة. ان نتائج الطرق الحدسية المقترحة تم مقارنتها مع بعض الطرق التي تعطي حلول دقيقة مثل طريقة العد التام (CEM) ومن ثم تم مقارنة نتائج تلك الطرق المقترحة مع بعضها لتحديد اي الطرق أكثر كفاءة.

INTRODUCTION

The Machine Scheduling Problem (MSP) has a number of recognized exact and approximation solution methods: Branch and Bound (BAB) method, Dynamic Programming (DP), and Complete Enumeration Method (CEM), are used to find exact solutions. [1].

Multi-criteria optimization is based on conflicting objective functions, resulting in a set of Pareto optimum solutions (or Efficient solution), which is seen as a part of one optimal solution. According to the objective functions, this set contains one (or many) solution(s) for which no other solution(s) is (are) better than this (these) solution(s). For multi-criteria scheduling problems, there are two ways in the literature [1]: the hierarchical approach and the simultaneous approach.

The most important literature surveys in the recent five years are: several effective algorithms for solving these problems are proposed.

Abdul-Razaq and Ali (2014) [2] introduced one of the important evolving algorithms which an Artificial Neural Network (ANN) in order to solve the single MSP (SMSP) by minimize the bi-criteria functions (BCF) $1//(\sum C_j, \sum T_j)$, this technique used to find best efficient and optimal schedule.

Abdul-Razaq and Ali (2016) [3] solved the $1//(\sum C_j, R_L)$ problem, they use BAB with new upper and lower bound and then using some Local Search Methods (LSM's) to solve problems with high number of jobs.

Abdul-Razaq and Ali (2016) [1] Solving SMSP to minimize the MCF $1//(\sum C_j, \sum T_j)$, they proposed two LSMs (Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)) to solve the above Single MSP (SMSP) by finding the set which represent the set of all efficient solutions. The results are compared with results of BAB method and they gave accurate results.

Ali and Abdul-Kareem (2017) [4], found the set of efficient solution for a MCF for SMSP by minimize tri-objective functions$(T_{max}, V_{max}, \sum V_j)$, by using BAB. They suggested new heuristic methods, for large number of jobs and using the best heuristic one to calculate a new upper bound for BAB method.

Chachan and Hameed (2019) [5], tried to solve new multiobjective problem $1//(\sum(C_j + T_j + E_j + V_j)$, they propose using BAB method in order to solve the above problem, by suggesting more than one upper and lower bounds, also they can the dominance rules DR's) concepts to minimize the consuming time by minimizing the number of nodes in the search tree of BAB method.

Ali and Ahmed (2020) [6] introduced a multi-criteria objectives function (MCF) $1//(\sum C_j, R_L, T_{max})$ problem in SMSP which is solved by BAB and some heuristic methods. Some special cases are introduced and proved to find the efficient solutions for problem. In [7] they solved $1//(\sum C_j + R_L + T_{max})$ problem to find good or optimal solutions by using exact and heuristic methods. Lastly, Bees Algorithm (BA) and PSO are used for solving the two suggested problems [8].

Khalaf (2021) [9], a hyper-heuristic method was proposed to incorporate the behavior of three optimized algorithms from the Bat algorithm (BAT). The method is based on the distribution of a specific implementation probability for each used algorithm and then updating this probability iteratively according to the results of each algorithm, and then we use random selection to determine the algorithm used in the current iteration.

Ibrahim et al. (2022) [10] they consider $1//(\sum(E_j + T_j + C_j + U_j + V_j)$ problem, to find a sequence that minimizes this MOF. They propose a BAB method to solve this problem. In addition, they use fast LSM's yielding near optimal solution. They report on computation experience; the performances of exact and LSM's are tested on large class of test problems.

In second section, the basic concepts for MSP are introduced. Third section discussed the most important methods of exact solutions for MSP. The mathematical formulation of BCF $1//(\sum T_j, R_L)$ problem and its special case (MOF $(\sum T_j + R_L)$) are given in forth section. In fifth section, we present BAB methods with revised lower and upper bounds as an exact method for solving the two problems. We present two heuristic techniques for the two problems in sixth section. The seventh section introduces the practical results and compare these outcomes with each other's. In eighth section, we will present the results of analysis of tables results and discussion for the data presented in the seventh section. Finally, the most relevant results and recommendations are presented in the ninth section.

Basic Concepts for MSP [1, 3]

Some Important Notations

First, we have to introduce the following notations:

n	:	The number of jobs.
p_j	:	The processing time of jobs j.
d_j	:	The Due date of jobs j.
C_j	:	The Completion time of job j, $C_j = \sum_{k=1}^{j} p_k$.
L_j	:	The Lateness time of job j, $L_j = C_j - d_j$.
T_j	:	The Tardiness time of job j, $T_j = max\{L_j, 0\}$.
$\sum T_j$	:	Total Tardiness time.
T_{max}	:	Maximum T_j, s.t. $T_{max} = max\{T_j\}$
L_{max}	:	Maximum L_j, $s.t.\ L_{max} = max\{L_j\}$
L_{min}	:	Minimum L_j, $s.t.\ L_{min} = min\{L_j\}$
R_L	:	The Range of lateness, s.t. $L_{max} - L_{min}$.

Basic Rules for MSP

Now we will discuss some basic and important definitions related to MSP.

***Definition* (1): Earliest Due Date (EDD) rule** [11]**:** Jobs are sequenced in non-decreasing order of due date (d_j), rule is used to minimize the problem $1//T_{max}$.

***Definition* (2)** [12]**:** In a multi-criteria decision-making dilemma, the phrase "**optimize**" refers to a solution in which there is no way to develop or improve one objective without deteriorating the other.

***Definition* (3)** [13]**:** If we cannot discover another schedule S' satisfying $f_j(S') \le f_j(S),\ j = 1, \dots, k,$

ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 33, Issue 3, 2022 DOI: http://doi.org/10.23851/mjs.v33i3.1135

with at least one of the above holding as a strict inequality, we call it an efficient schedule. Another way of putting it is that S' dominates S.

Dominance Rules in MSP

Dominance Rules (DRs) can usually identify some (or all) sections of the permutation or sequence in order to acquire the best value for the problem's objective function. The SRs can help evaluate whether a node in the BAB method can be disregarded or deleted without calculating its LB. Within BAB, the DR's are also beneficial for canceling all the nodes that are controlled by others. These enhancements suggest a significant reduction in the number of nodes required to get the optimal solution for the problem [1].

***Definition* (4) [13]:** If G is a graph with n vertices, the **matrix** $A(G) = [a_{ij}]$, $i,j = 1,2,\dots,n$, whose $i^{th} and\ j^{th}$ member is 1 if there is at least one edge or path between vertex V_i and vertex V_j, and zero otherwise, is termed the **adjacency matrix** of graph G, where:

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

***Remark (1)* [1]:** For $1//\sum T_j$ problem if $p_i \le p_j$ and $d_i \le d_j$, then we can obtain an optimal solution in which job j sequenced after job i.

Remark (2): EDD rule may be useful for $1//\sum T_j$ problem to obtain optimal solution.

Exact Solving Methods for MSP

Branch and Bound Method [1]

Branch and bound (BAB) methods are implicit enumeration techniques, which can find an optimal solution by systematically examining subsets of feasible solutions. These methods are usually described by means of search tree with nodes that corresponding to these subsets.

Dynamic Programming [14]

Dynamic programming (DP) is the general approach to making a series of connected decisions optimally. DP determines the optimal solution to a multivariate problem by dividing the problem into stages, with each stage having a sub-problem that aims to find the optimal value for only one variable. The characteristic feature is due to dealing with only one variable and is much easier mathematically than dealing with all variables at the same time.

Mathematical Description for the Suggested Problem

Let us have a set of n jobs $N = \{1,2,\dots,n\}$ on a SMSP. To obtain the suitable mathematical formulation for our problem we have to search about $\sigma \in S$ (S can be considered as the set of total feasible sequences). σ can be used to minimize the BCF $(\sum T_j, R_L)$.This problem can be written as:

$$Min\{\sum T_j, R_L\}$$

Subject to

$$\left.\begin{array}{ll} C_j \ge p_{\sigma(j)}, & j = 1,2,\dots,n. \\ C_j = C_{(j-1)} + p_{\sigma(j)}, & j = 2,3,\dots,n. \\ L_j = C_j - d_{\sigma(j)}, & j = 1,2,\dots,n. \\ T_j \ge C_j - d_{\sigma(j)}, & j = 1,2,\dots,n. \\ R_L = L_{max} - L_{min} & \\ T_j, R_L \ge 0, & j = 1,2,\dots,n. \end{array}\right\} \quad (TR)$$

Problem (TR) (is called TR according to objectives $\sum T_j$ and R_L) is NP-hard since the two criteria are NP-hard problems.

For *SP*-problem, we can deduce sub-problem: The $1//\sum T_j + R_L$ Problem:

$$Min\{\sum T_j + R_L\}$$

Subject to

$$\left.\begin{array}{ll} C_j \ge p_{\sigma(j)}, & j = 1,2,\dots,n. \\ C_j = C_{(j-1)} + p_{\sigma(j)}, & j = 2,3,\dots,n. \\ L_j = C_j - d_{\sigma(j)}, & j = 1,2,\dots,n \\ T_j \ge C_j - d_{\sigma(j)}, & j = 1,2,\dots,n. \\ R_L = L_{max} - L_{min} & \\ T_j, R_L \ge 0, & j = 1,2,\dots,n. \end{array}\right\} \quad (STR)$$

This subproblem is called Problem (*STR*).

Example (1): Suppose we have the following table for $n = 4$:

n	**1**	**2**	**3**	**4**
p_j	7	3	5	6
d_j	20	9	12	6

By remark (1), we obtain the following DR: $2 \rightarrow 1, 3 \rightarrow 1, 4 \rightarrow 1, 2 \rightarrow 3$. Then the adjacency matrix:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & a_{24} \\ 1 & 0 & 0 & a_{34} \\ 1 & \bar{a}_{24} & \bar{a}_{34} & 0 \end{bmatrix}$$

Here we obtain three sequences the best one is $\pi = (4,2,3,1)$ which gives $\left(\sum_{j=1}^{n} T_j, R_L\right) = (3,2)$, as shown in Table 1:

Table 1. Calculating the objective functions of example (1)

n	**4**	**2**	**3**	**1**
p_j	6	3	5	7
d_j	6	9	12	20
C_j	6	9	14	21
L_j	0	0	2	1
T_j	0	0	2	1

Exact Methods for TR and STR-Problems

In this part of paper, some exact methods for the BCF and BOF problems are discussed, the exact methods are BAB methods one for MCF and the second for BOF.

Finding the Exact Solution(s) for TR-Problems

In this section we suggest using BAB algorithm with a few tweaks, which we call it BAB(TR). The branching mechanism, lower and upper bounding procedures, and search technique all distinguish the BAB method.

When the two criteria $\sum T_j$ and R_L are of simultaneous relevance in the issue, we provide a constructive BAB (TR) to identify **Pareto optimal points (PP)**) of problem (TR) to obtain some or all the efficient solutions. The main idea behind this BAB algorithm is to combine the benefits of the BAB(TR) technique with a few tweaks, such as using efficient solution definitions and not resetting the upper limit (UB) at the final level of BAB(TR).

Before we discuss the BAB(TR) procedure, we have to discuss the derivation of $UB \text{ and } LB$.

For the UB we suggest to find UB by $\sigma = EDD$, and set $UB = \left(\sum T_{\sigma(j)}, R_{L\sigma(j)}\right)$ and the set of efficient solutions $S = \{\sigma\}$. While for LB we suggest using EDD rule of unsequenced jobs.

The suggested BAB algorithm can be shown as follows:

BAB(TR) algorithm

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

Step (1): Find the sets UB and S.

Step (2): For search tree, for each node, calculate the LB (α) s.t. LB (α)= cost of branched nodes + cost of S which represents the unsequenced jobs, this done by using EDD rule for the usequenced part.

Step (3): When the $LB(\alpha)$ UB we branch from current node.

Step (4): In the last level (n) of BAB tree, where $\left(\sum T_j, R_L\right)$ denote the outcome, this result is added to the set of PP, if it is not dominated by the previously obtained PP.

Step (5): **STOP**.

Finding the Exact Solution for STR-Problems

For STR-Problem we can use the same BAB(TR) which is used for P-problem, of course with some modification which we denoted by BAB (STR). For UB, the same UB for STR-problem s.t. $UB(\sigma = EDD) = \left(\sum T_{\sigma(j)} + R_{L\sigma(j)}\right)$.and so on for LB s.t. $LB(\pi = EDD) = \left(\sum T_{\pi(j)} + R_{L\mu(j)}\right)$, where π is the EDD rule for unsequenced jobs.

New Approximation Methods for Solving TR and STR-Problems

Because the majority of scheduling issues are NP-hard, and solving them using BAB or DP approaches could take a long time, that make many research academics to created approximation or heuristic algorithms to solve them quickly and effectively.

According to Reeves [15], the heuristic (or approximation) strategy is as follows: A heuristic is a strategy or algorithm that searches for optimal or near-optimal answers in a reasonable amount of time with no guarantee of optimality, or even to check how close this solution is to an optimal solution in many circumstances.

For TR and STR-problems, we will suggest new approximation methods for the two problems, these methods are discussed in the following subsections.

New Approximation Methods for Solving TR-Problem

The First Approximation Method for TR-Problem

EDD is required for the first heuristic technique. Because the EDD rule solves the $1//\sum T_j$ problem well, we recommend first using the EDD rule to order the jobs, then computing the value of the objective function, after that make the second job in the first place, and so on until you have n sequences. The process repeated for EDD rule for

ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 33, Issue 3, 2022 DOI: http://doi.org/10.23851/mjs.v33i3.1135

TR-problem (so we called it EDD-STRL(TR)). The algorithm of EDD-STRL(TR) is as follows:

EDD_STRL(TR) Heuristic Algorithm

Step (1):INPUT: n, p_j and $d_j, j = 1,2,3, \dots, n,\ S = \phi$.

Step (2): Sorting the jobs in EDD rule (non-decreasing order) to obtain (σ_1), and compute the MCF

$$F_{11}(\pi_1) = \left(\sum T_j(\pi_1), R_L(\pi_1)\right);$$

$$S = S \cup \{F_{11}(\pi_1)\}.$$

Step (3): FOR $i = 2, \dots, n$

Put the job i in the first position of the schedule σ_{i-1} to obtain new schedule σ_i then calculate $F_{1i}(\pi_i) = \left(\sum T_j(\pi_1), R_L(\pi_1)\right);$

$$S = S \cup \{F_{1i}(\pi_i)\}.$$

END;

Step (4): Now we will filtering the set S to gain only the set of efficient solutions for P-problem by.

Step (5): OUTPUT The set S.

Step (6): STOP.

The Second Approximation Method for TR-Problem

The other heuristic method is based on the DRs stated in remark (1), and it is summarized by choosing a sequence with the smallest p_j and d_j that does not contradict the problem's DR, and then computing the objective function. The DR-STRL(TR) algorithms can be summarized in following steps:

DR_STRL (TR) Heuristic Algorithm

Step (1): INPUT: n, p_j and $d_j, j = 1,2, \dots, n$.

Step (2): Apply Remark (1), to gain the DR's and the corresponding adjacency matrix A;

$$\pi = \varphi, N = \{1,2, \dots, n\}, S = \varphi.$$

Step (3): sort the jobs in non-increasing order of p_j to gain π_1 which be accepted under condition that it's not contradiction with matrix A, if there exist more than one job order π_1 by d_j, then $S = S \cup \{\pi_1\}$.

Step (4): sort the jobs in non-increasing order of d_j to gain π_2 which be accepted under condition that it's not contradiction with matrix A, if there exist more than one job order π_2 by d_j, then $S = S \cup \{\pi_2\}$.

Step (5): Find the dominated set S' from S.

Step (6): Compute the BCF $F(S')$.

Step (7): OUTPUT The set S' which is considered a set of efficient solutions.

Step (8): STOP.

New Approximation Methods for Solving STR-Problem

In this section we propose to use the EDD-STRL(TR) and DR-STRL(TR) which are suggested in previous section for TR-problem.

Practical Examples for using the Suggested Methods

In this section we will generate (5) examples generated randomly for p_j *and* d_j for s.t. $p_j \in [1,10]$ and

$$d_j \in \begin{cases} [1,30], & 1 \le n \le 29. \\ [1,40], & 30 \le n \le 99. \\ [1,50], & 100 \le n \le 999. \\ [1,70], & \text{otherwise.} \end{cases}$$

under condition that $d_j \ge p_j$, for $j = 1,2, \dots, n$.

First let's defined the following abbreviations (are sorted alphabetically):

AAE:	Average Absolute Error.
$ABCF$:	Average Bi-Criteria Function.
$ABOF$:	Average Bi-Objective Function.
ANS:	Average number of efficient solutions.
AT:	Average of Time per second.
AV:	Average.
En:	Example Number.
Re:	$0 < Real < 1$.

Comparison Results of TR-Problem

Comparison of the efficient results between BAB(TR) and CEM(TR) for TR-problem are shown in Table 2, $n = 5{:}\,2{:}\,11$.

Table 2. Comparison results between BAB (TR) and CEM (TR) for TR-problem, for $n = 5{:}\,2{:}\,11$.

n	CEM (TR)			BAB (TR)			
	$ABCF$	AT	ANS	$ABCF$	AT	ANS	AAE
5	(10.7,12.3)	Re	1.2	(10.7,12.3)	Re	1.2	(0,0)
7	(34.1,18.9)	Re	2.2	(34.1,18.9)	Re	2.2	(0,0)
9	(108.7,37.7)	3.7	4.4	(107.5,38.3)	Re	4.2	(2.1,0.6)
11	(143.0,44.3)	539.9	4.0	(152.0,45.8)	Re	2.0	(9.4,1.5)
Av	(74.1,28.3)	135.9	3.0	(76.0,28.8)	Re	2.4	(2.9,0.5)

Comparison results of BAB(TR) with each EDD-STRL(TR) and DR-STRL(TR) for the efficient solutions results for TR-problem are shown in table 3, for $n = 5{:}\,5{:}\,40$.

For BAB(TR), $AT = Re$ for $n = 5{:}\,30$, while $AT = 101.3, 103.1$ for $n = 35$ and 40 and $Av = 25.5$ for. For SR-STRL(TR), $ANS = 2.0$. for all n.

Table 3. Comparison between EDD-STRL(TR) and DR-STRL(TR) with CEM(TR) for TR-problem, $n = 5{:}\,5{:}\,40$

n	BAB (TR)		EDD-STRL (TR)		DR-STRL (TR)
	$ABCF$	ANS	ANS	$ABCF$	$ABCF$
5	(10.7,12.3)	1.2	1.2	(12.3,19.0)	(11.6,15.8)
10	(79.8,30.3)	3.2	2.0	(77.1,43.5)	(91.5,39.8)
15	(252.1,55.8)	4.2	4.4	(247.4,77.5)	(287.7,70.3)
20	(540.1,82.7)	5.0	5.0	(508.6,99.0)	(603.5,91.7)
25	(1036.5,120.5)	4.6	5.4	(1000.8,139.9)	(1216.9,128.4)
30	(1442.1,134.7)	4.4	5.6	(1371.7,167.3)	(1627.0,149.8)
35	(2111.5,174.0)	5.8	5.8	(2073.4,201.7)	(2456.3,184.9)
40	(2516.4,190.2)	3.8	6.4	(2474.8,222.5)	(2991.5,201.3)
Av	(998.7,100.1)	4.0	4.5	(970.8,121.3)	(1160.8,110.3)

Notice that the Heuristics EDD-STRL(TR) and DR-STRL(TR) give good results compared with BAB(TR) for TR-problem.

Table 4 shows the comparison results between the two suggested heuristic methods EDD-STRL(TR) and DR-STRL(TR) to obtain an efficient solution for TR-problem for $n = 30,70,100,300,700, 1000\ and\ 3000$.

Table 4. A comparison results between EDD-STRL(TR) and DR-STRL(TR) for different n.

n	EDD-STRL(TR)			DR-STRL(TR)		
	$ABCF$	AT	**ANS**	$ABCF$	AT	**ANS**
50	(3963.6,266.5)	Re	8.6	(4752.7,246.9)	Re	2.0
80	(11467.9,457.9)	Re	8.4	(13868.5,428.7)	Re	2.0
100	(17525.4,573.4)	Re	8.8	(21355.1,534.2)	Re	2.0
300	(167654.3,1661.1)	Re	9.8	(205087.8,1614.8)	Re	2.0
700	(922776.8,3853.2)	1.7	10.4	(1040337.4,3796.4)	Re	1.6
1000	(1895637.8,5515.8)	3.7	10.6	(2210376.4,5430.3)	1.5	1.8
3000	(17190780.6,16504.4)	103.2	11.0	(17184480.4,16418.4)	35.3	1.0
Av	(2887115.2,4118.9)	15.5	0.7	(2954322.6,4067.1)	5.3	1.8

ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 33, Issue 3, 2022 DOI: http://doi.org/10.23851/mjs.v33i3.1135

Comparison Results of STR-problem

In table 5 we will show some of Comparison results between BAB(STR) and CEM(STR) for *STR*-problem, for $n = 5{:}\,2{:}\,11$.

In table 6 we show the results of BAB(STR) with the results of the two suggested heuristics methods EDD-STRL(STR) and DR-STRL(STR), for $n = 3{:}\,2{:}\,11$, for *STR*-problem.

Table 5. Comparison between BAB(STR) and CEM(STR) for *STR*-problem, for $n = 5{:}\,2{:}\,11$.

n	CEM(STR)		BAB (STR)	
	$ABCF$	AT	$ABCF$	AT
5	22.8	Re	22.8	Re
7	51.0	Re	51.0	Re
9	141.6	3.8	141.6	Re
11	169.0	764.2	169.0	233.9
Av	**99.5**	**192.0**	**99.5**	**58.5**

Table 6. Comparison between BAB(STR), EDD-STRL(STR) and DR-STRL(STR) $n = 3{:}\,2{:}\,11$, for *STR*-problem.

n	BAB (STR)		EDD-STRL (STR)			DR-STRL (STR)		
	OP	AT	$ABOF$	AT	AAE	$ABOF$	AT	AAE
3	12.2	Re	14.0	Re	1.8	12.2	Re	0.0
5	22.8	Re	27.0	Re	4.2	24.0	Re	1.2
7	51.0	Re	64.8	Re	13.8	61.8	Re	10.8
9	141.6	Re	156.4	Re	14.8	156.8	Re	15.2
11	169.0	233.9	202.0	Re	33.0	195.8	Re	26.8
Av	79.3	46.8	92.8	Re	13.5	90.1	Re	10.8

Notice that the heuristics EDD-STRL(STR) and DR-STRL(STR) give near objective values compared with BAB(STR), and that can be noticed from AAE, for *STR*-problem.

Figure 1 shows the curves of the results of the suggested three methods: BAB(STR), EDD-STRL(STR) and DR-STRL(STR), for *STR*-problem, for $n = 3{:}\,2{:}\,11$ (see Table 6).

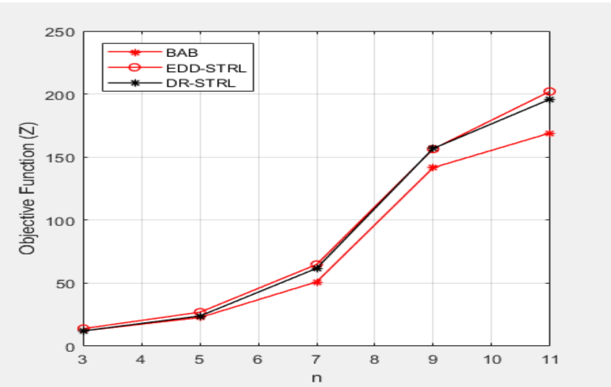

Figure 1. Lines curves results of BAB(STR), EDD-STRL(STR) and DR-STRL(STR) for $n = 3{:}\,2{:}\,11$.

Table 7 describes the average of the best solutions for STR-problem for $n =$ 30, 70, 100, 300, 700, 1000 and 2000 of the two suggested EDD-STRL(STR) and DR-STRL(STR).

Table 7. Results of averages of results of EDD-STRL(STR) and DR-STRL(STR) for STR-problem, $n = 30,70,100,300,700,1000$ *and* 3000.

n	EDD-STRL (STR)		DR-STRL (STR)	
	$ABOF$	AT	$ABOF$	AT
30	1514.4	Re	1511.2	Re
70	8564.4	Re	8536.6	Re
100	17981.2	Re	17930.8	Re
300	168865.2	Re	168750.0	Re
700	925509.0	1.6	925117.2	Re
1000	1899512.8	2.9	1898722.0	Re
3000	17201690.8	23.5	17200898.8	25.8
Av	2889091.1	4.0	2888780.9	3.7

Analysis the Results with Discussion for the Two Problems

For *TR*-problem:

1. We notice the accuracy of the results of BAB(TR) comparing with results of CEM(TR) (see Table 2).
2. For accuracy, the EDD-STRL(TR) is the best among DR-STRL(TR) and BAB(TR) for the function $\sum T_j$ while BAB(TR) is the best for the function R_L and in CPU-time for $n \leq 40$ (see Table 3).
3. For $n \leq 3000$, EDD-STRL(TR) is the better from DR-STRL(TR) for the function $\sum T_j$ while DR-STRL(TR) is better from EDD-STRL(TR) for function R_L and in CPU-time (see Table 4).

For *STR*-problem:

1. From Table 5, we notice the accuracy of results of BAB(STR) comparing with the results of CEM(STR) for $AAE = 0$ for all $n \leq 11$.
2. By comparing the results of the suggested heuristic methods (EDD-STRL(STR) and DR-STRL(STR)) with BAB(STR), we notice BAB(STR) is the best (see Table 6).
3. Lastly, from Table 7, the results accuracy of DR-STRL(STR) is better from the results of EDD-STRL(STR) for $n \leq 3000$.

CONCLUSIONS AND FUTURE WORK

1. We discussing the problems The BCF $1//(\sum T_j, R_L)$ and the BOF $1//(\sum T_j + R_L)$, and find the mathematical formulation for them.
2. We suggesting using BAB method for the two problems, which is considered as an exact method. The results prove the accuracy of BAB results.
3. We proposing two new heuristics: EDD-STRL and DR-STRL with good performance for the two discussed problems.
4. As seen in remark (1), the $\sum T_j$ function can be solved depending on some DR's, as future work, these DR's can be useful when using BAB method to solve the two problems with high number of jobs in reasonable time.
5. We suggest using new LSM for solving TR and STR-problems, like Simulated Annealing (SA), Coco Algorithm (CA), and Ant Colony Optimization (ACO).

ACKNOWLEDGMENTS

The authors would like to thanks Mustansiriyah University at (www.uomustansiriyah.edu.iq) Baghdad – Iraq for its support in the present work.

REFERENCES

[1] F. 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.

[2] T. S. Abdul-Razaq and F. H. Ali, "Constructing of an artificial neural network to minimize total completion time and total tardiness", IOSR Journal of Mathematics (IOSR-JM) e-ISSN: 2278-3008, p-ISSN: 2319-7676. Volume 10, Issue 2 Ver. VI (Mar-Apr. 2014), PP 25-37. https://doi.org/10.9790/5728-10262537

[3] T. S. Abdul-Razaq and S. A. Ali, "A comparison of local search algorithm for multi-criteria scheduling problems", M.Sc. thesis, University of Al-Mustansiriyah Univ, College of Science, Dept. of Mathematics, 2016.

[4] S. B. Abdulkareem, "Improving exact and local search algorithms for solving some combinatorial optimization problems", M. Sc., Thesis, Mustansiriyah University, College of Science, Dept. of Mathematics, 2017.

[5] H. A. Chachan and A. S. Hameed, "Exact methods for solving multi-objective problem on single machine scheduling", Iraqi Journal of Science, 2019, Vol. 60, No.8, pp:1802-1813. https://doi.org/10.24996/ijs.2019.60.8.17

[6] F. H. Ali and M. G. Ahmed, "Efficient algorithms to solve tricriteria machine scheduling problem", (15th and the second International) Conference of Statistical Applications (ICSA2020), Irbil, Kurdistan Region-Iraq, 12-13/Feb./2020, Journal of Al Rafidain University College, Volume, Issue 46, pp: 485-493, 2020. https://www.iasj.net/iasj/article/190968.

[7] F. H. Ali and M. G. Ahmed, "Optimal and near optimal solutions for multi objective function on a single machine", 1st International Conference on Computer Science and Software Engineering (CSASE2020), Duhok, Kurdistan Region-Iraq, Sponsored by IEEE Iraq section, 16-17/Apr./2020. https://doi.org/10.1109/CSASE48920.2020.9142053

[8] F. H. Ali and M. G. Ahmed, "Local search methods for solving total completion times, range of lateness and maximum tardiness problem", 6th International Engineering Conference (IEC2020), Irbil, Kurdistan Region-Iraq, Sponsored by IEEE Iraq section, 26-27/Feb./2020. https://doi.org/10.1109/IEC49899.2020.9122821

[9] W. S. Khalaf, "Ensemble bat algorithm based on hyper heuristic approach for solving unconstrained optimization problems", Turkish Journal of Computer

and Mathematics EducationVol.12 No.10, pp:5466-5478, 2021, https://doi.org/10.17762/turcomat.v12i10.5352

[10] M. H. Ibrahim, F. A. Ali and H. A. Chachan, "Solving multi-objectives function problem using branch and bound and local search methods", International Journal of Nonlinear and Applications (IJNAA), (Scopus) ISSN 2008-6822, 13 (2022) No. 1, 1649-1658. http://dx.doi.org/10.22075/ijnaa. 2022.5780

[11] J. A. Hoogeveen, "Minimizing maximum earliness and maximum lateness on a single machine", Center for Mathematics and Computer science, P.O. Box 4079, 1009 AB Amsterdam, The Netherland, https://dl.acm.org/doi/10.5555/645585.659304

[12] J. A. Hoogeveen, "Single machine scheduling to minimize a function of two or three maximum cost criteria", Journal of Algorithms, 21, 415-433, 1996. https://doi.org/10.1006/jagm.1996.0051

[13] B. Kolman, Introductory linear algebra with applications, Macmillan Publishing Company, 1988.

[14] W. S. Khalaf, M. M. Shakir and N. Y. Abd-Alredaa "Maximizing the performance of the Iraqi Armed Forces and determining the optimal path for them using the dynamic programming", Int. J. Nonlinear Anal. Appl. Volume 12, Special Issue, Winter and Spring 2021, pp: 847-860 ISSN: 2008-6822 (electronic) http://dx.doi.org/10.22075/ijnaa.2021.5465

[15] C. R. Reeves, Modern Heuristic Techniques for Combinatorial Problems, John Wiley and sons, New York, 1993.

How to Cite

F. H. Ali, R. N. . Jawad, and W. A. . Hussein, “Solving Bi-Criteria and Bi-Objectives of Total Tardiness Jobs Times and Range of Lateness Problems Using New Techniques”, *Al-Mustansiriyah Journal of Science*, vol. 33, no. 3, pp. 27–35, 2022.