Al-Mustansiriyah Journal of Science
ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 34, Issue 4, 2023 DOI: https://doi.org/10.23851/mjs.v34i4.1430

Research Article Open Access

Using Local Search Methods for Solving Two Multi-Criteria Machine Scheduling Problems

Safanah Faisal Yousif[1*], Faez Hassan Ali[1], Karrar Fatah Alshaikhli[2]

[1] Department of Mathematics, College of Science, Mustansiriyah University, 10052 Baghdad, IRAQ.
[2] Department of Engineering and Mathematics, St. Philip's College, San Antonio, Texas, USA.

*Correspondent contact: safanafaysal77@gmail.com

Article Info

Received
12/06/2023

Revised
23/06/2023
22/07/2023

Accepted
31/07/2023

Published
30/12/2023

ABSTRACT

In this paper, we have improved solutions for two of the Multi-Criteria Machine Scheduling Problems (MCMSP). These problems are to maximize early jobs time and range of lateness jobs times $1//(E_{max}, R_L)$), and the second problem is maximum tardy jobs time and range of lateness jobs times $1//(T_{max}, R_L)$ in a single machine with Multi-Objective Machine Scheduling Problems (MOMSP) $1//(E_{max}, R_L)$and $1//(T_{max}, R_L)$ which are derived from the main problems respectively. The Local Search Methods (LSMs), Bees Algorithm (BA), and a Simulated Annealing (SA) are applied to solve all suggested problems. Finally, the experimental results of the LSMs are compared with the results of the Branch and Bound (BAB) method for a reasonable time. These results are ensuring the efficiency of LSMs.

KEYWORDS: Local search Methods, Multi-Criteria Scheduling Problems, Bees Algorithm, Simulated Annealing, Branch and Bound Method.

الخلاصة

في هذا البحث، تم تحسين حل مسألتين من مسائل جدولة الماكنة متعددة المعايير. المسالة الاولى هي مسالة القيمة العظمى لوقت الاعمال المبكرة ومدى زمن الاعمال المتأخرة $1//(E_{max}, R_L)$ مسألة الثانية هي مسالة القيمة العظمى لزمن الاعمال ذات التاخير اللاسلبي ومدى زمن الاعمال المتأخرة $1//(T_{max}, R_L)$ على ماكنة واحدة، مع مسائل جدولة الماكنة متعددة الاهداف $1//(E_{max}, R_L)$ و $1//(T_{max}, R_L)$ المشتقة من المسالتين الرئيسية على التوالي. تم تطبيق نوعين من طرق البحث المحلي (LSMs) وهي خوارزمية النحل (BA) ومحاكاة التلدين (SA) لحل جميع المسائل المقترحة. واخيرا تم مقارنة النتائج العملية بين طرق البحث المحلي مع نتائج طريقة BAB ضمن الوقت المقبول، النتائج اثبتت كفاءة طرق البحث المحلية المقترحة.

INTRODUCTION

It is widely acknowledged that one of the key areas of combinatorial optimization problems is the machine Scheduling Problems (MSP). Finding the best and most advantageous solution for minimizing a function is the main goal of this paper. The computational time requirements are very high for large-sized problems when the MSP is an NP-hard problem. We can make use of heuristic techniques to get around these drawbacks. The development of heuristic methods has been referred to as "local search methods" in recent years. In a reasonable amount of time, the local search approach offers high-quality solutions to NP-hard problems of realistic size. Beginning with a first solution, the Local Search Methods (LSMs) keep looking for better ones by searching neighborhoods [1]. In 2014, Ibraheem [2], investigated the $1//T_{max} + E_{max}$ problem and found a near optimal solution by using LSMs (Descent method (DM), and Simulated Annealing (SA) algorithm), respectively. Also, the study reported the results of extensive computational tests of (DM), (SA). Our experimental results indicate that the proposed algorithms have found exact and efficient solutions in most cases. In 2014, Mahmood [3] discussed the multi-criteria scheduling problem and studied on a single machine to find efficient solutions of the problem, $1//F(\sum C_j, T_{max}, V_{\max})$, and used the LSMs to find approximation solutions. She suggested LSMs for finding approximation solutions. These LSMs are: (DM), (SA) and a Genetic Algorithm (GA) which are implemented. Based on the results of computational experiments, conclusions are formulated on the efficiency of the local search algorithms. In 2017, Abdulkareem [4] presented the problem $(T_{max}, V_{max}, \sum V_j)$, and solved it by using some types of local search methods: Particle Swarm Optimization (PSO) and Bees Algorithm (BA), and neural networks (NN) are used to solve the origin problem. Abbas (2019) [5] studied multi-objective single MSP. The objective is to minimize four cost functions $(\sum C_j + \sum U_j + \sum T_j + T_{max})$ by LSMs method and provide solutions for the considered problem. In 2022, Ahmed [6] offered some methods to solve the MCMSP by minimizing $(1//(\sum C_j, T_{max}, R_L))$. She deduced sub-problems denoted by $(1//(\sum C_j + T_{max} + R_L))$ suggested (8) solving methods classified as exact, heuristic and local search methods to find the set of efficient, optimal, near optimal and approximate solutions for the two problems. The rest of the paper is organized as follows: In Section 2, we discuss the MSP concept. In Section 3, we introduce the mathematical formulation of the BCMSP and BOMSP. Also, we revisit the local search method (LSMs); Bees Algorithm (BA) and a Simulated Annealing (SA). In Section 4. We apply the LSMs for solving the two BCMSP and BOMSP in Sections 5 and 6, respectively. The analysis and discussion of the comparison results are discussed in Section 7. Conclusions and recommendations are introduced in Section 8.

MATERIALS AND METHODS

Machine Scheduling Problem Concept

In this Section, we start by introducing some important notations where we concentrate on the performance criteria without elaborating on the machine environment. It is assumed that there are n jobs, which we denoted by $1, \dots, n$ and these jobs are to be scheduled on a set of machines that are continuously available from time zero onwards and can handle only one job at a time. We only state here that the notations are used for single machine, jobs $j, (j = 1, \dots, n)$ has [7]:

p_j : which mean that the job j has to processed for a period of length p_j.

d_j: a due date, the date when the job j should be completed. Even though it is permitted, there is a cost associated with finishing the task after the deadline. It is known as a deadline when the due date absolutely must be met, and it is known as a common due date when it applies to all jobs.

s_j: a slack time of job j s.t. $s_j = d_j - p_j$.

C_j : the completion time, the time at which the processing of job j is completed s.t. $C_j = \sum_{k=1}^{j} p_k$.

Now for a given sequence σ of jobs we can compute for job j:

- The lateness $L_j = C_j - d_j$
- Range of lateness: $R_L = L_{max} - L_{min}$ where $L_{max} = \max_{1\le j\le n}\{L_j\}$ and $L_{min} = \min_{1\le j\le n}\{L_j\}$.
- The tardiness $T_j = \max\{L_j, 0\}$.
- The maximum tardiness: $T_{max} = \max_{1\le j\le n}(T_j)$.
- The earliness $E_j = max\{-L_j, 0\}$.
- The maximum earliness: $E_{max} = \max_{1\le j\le n}\{E_j\}$.

The following sequencing rules and basic concepts are used in this work:

Definition (1): (Earliest Due Date (EDD) rule [7]**):** Sequencing the jobs in non-decreasing order of their due dates (d_j) i.e., $(d_1 \le d_2 \le \cdots \le d_n)$, which is solving the problems $1//L_{max}$ and $1//T_{max}$.

Definition (2): (Minimum slack Time (MST) rule [6]**):** Jobs are sequenced in non- decreasing order of slack time (s_j) i.e., $(s_1 \le s_2 \le \cdots \le s_n)$. This rule is well known for solving the problem $1//E_{max}$

Mathematical Formulation of the Bcmsp and Bomsp

Let $N = \{1,2, \dots, n\}$ be a set of jobs that is wanted to be scheduled on a BCMSP with $p_j \le d_j$ and BOMSP for each NP-hard problem. The MSP can process only one job at a time using the two field's classification. In the following two subsections we will discuss the mathematical formulation of BCMSP and BOMSP for each problem.

Mathematical Formulation for First Problem

This MCMSP denoted by $1//(E_{max}, R_L)$ which can be formulated for a given schedule $\sigma = (1,2, \dots, n)$ as:

$F = \min(E_{max}, R_L)$
Such that

$$\left.\begin{array}{ll} C_1 = p_{\sigma_{(1)}} & \\ C_j \geq p_{\sigma_{(j)}}, & j = 1,2,\dots,n \\ C_j = C_{j-1} + p_{\sigma_{(j)}}, & j = 2,3,\dots,n \\ L_j = C_j - d_{\sigma_{(j)}}, & j = 1,2,\dots,n \\ R_{L(\sigma)} = L_{\max(\sigma)} - L_{\min(\sigma)} & \\ E_j \geq d_{\sigma(j)} - C_j\,, & j = 1,2,\dots,n \\ E_{max} = \max\limits_{1\leq j\leq n}\{E_j\} & \\ E_j \geq 0, & j = 1,2,\dots,n \\ E_{max}(\sigma), R_L(\sigma) \geq 0 & \end{array}\right\} \dots(ER)$$

while the MOMSP of the MCMSP is denoted by $1//(E_{max}, R_L)$ which is formulated as follow for the BOMSP using schedule $\sigma = (1,2,\dots,n)$ as:

$$F = \min(E_{max} + R_L)$$

Such that

$$\left.\begin{array}{ll} C_1 = p_{\sigma_{(1)}} & \\ C_j \geq p_{\sigma_{(j)}}, & j = 1,2,\dots,n \\ C_j = C_{j-1} + p_{\sigma_{(j)}}, & j = 2,3,\dots,n \\ L_j = C_j - d_{\sigma_{(j)}}, & j = 1,2,\dots,n \\ R_{L(\sigma)} = L_{\max(\sigma)} - L_{\min(\sigma)} & \\ E_j \geq d_{\sigma(j)} - C_j\,, & j = 1,2,\dots,n \\ E_{max} = \max\limits_{1\leq j\leq n}\{E_j\} & \\ E_j \geq 0, & j = 1,2,\dots,n \\ E_{max}(\sigma), R_L(\sigma) \geq 0 & \end{array}\right\} \dots\text{(EPR)}$$

We see [8].

Mathematical Formulation for second Problem

The MCMSP is denoted by $1//(T_{max}, R_L)$, which can be formulated for a given schedule $\sigma = (1,2,\dots,n)$ as:

$F = \min(T_{max}, R_L)$
Such that

$$\left.\begin{array}{ll} C_j = \sum\limits_{k=1}^{j} p_k\,, & j = 1,2,\dots,n \\ C_j = C_{j-1} + p_{\sigma_{(j)}}, & j = 2,3,\dots,n \\ L_j = C_j - d_{\sigma_{(j)}}, & j = 1,2,\dots,n \\ T_j \geq C_j - d_{\sigma(j)}, & j = 1,2,\dots,n \\ T_{max} = \max\limits_{1\leq j\leq n}\{T_j\} & \\ R_{L(\sigma)} = L_{\max(\sigma)} - L_{\min(\sigma)} & \\ T_j \geq 0, & j = 1,2,\dots,n \\ T_{max}(\sigma), R_L(\sigma) \geq 0 & \end{array}\right\} \dots(TR)$$

while the MOMSP of the MCMSP is denoted by $1//(T_{max}, R_L)$ which is formulated as follows for the BOMSP using schedule $\sigma = (1,2,\dots,n)$ as:
$F = \min(T_{max} + R_L)$
such that

$$\left.\begin{array}{ll} C_1 = p_{\sigma_{(1)}} & \\ C_j \geq p_{\sigma_{(j)}}, & j = 1,2,\dots,n \\ C_j = \sum\limits_{k=1}^{j} p_k\,, & j = 1,2,\dots,n \\ C_j = C_{j-1} + p_{\sigma_{(j)}}, & j = 2,3,\dots,n \\ L_j = C_j - d_{\sigma_{(j)}}, & j = 1,2,\dots,n \\ R_{L(\sigma)} = L_{\max(\sigma)} - L_{\min(\sigma)} & \\ T_{max} = \max\limits_{1\leq j\leq n}\{T_j\} & \\ R_{L(\sigma)} = L_{\max(\sigma)} - L_{\min(\sigma)} & \\ T_j \geq 0, & j = 1,2,\dots,n \\ T_{max}(\sigma), R_L(\sigma) \geq 0. & \end{array}\right\} \dots(TPR)$$

For more details, readers may see [8].

Local Search Methods

In this Section, we will discuss some LSMs for two types of BCMSP and BOMSP problems and find near-optimal solutions for such problems in reasonable computational time to avoid solving problems that require large computational times [9].

Simulated Annealing

An optimization method based on trajectories is called Simulated Annealing (SA). It is essentially a strategy for continuous improvement with a criterion that occasionally accepts higher cost configurations. In the 80s century, SA was first used to resolve the COP. The physical annealing of materials, which involves first heating the solid and then gradually reducing it to a lower energy state, served as the inspiration for SA. Due to its ability to simulate how thermodynamic systems transition from one state to another, the Metropolis acceptance criterion is used to determine whether the current solution should be accepted or rejected. [10]. The initial state of a thermodynamic system was selected at energy (Cost or C) and temperature (Temperature (Temp)). The system's initial configuration is altered to produce a new configuration while holding t constant, and the change in energy ΔC is calculated. The new configuration is accepted without conditions, but if ΔC is negative whereas it is accepted if ΔC is positive with a probability given by the Boltzmann factor shown in (1) to avoid trapping in the local optima.

Algorithm (1): Simulated Annealing (SA)

Step1: Input: Temp., Final Temp., cooling rate, ch;
Step2: $ch' = ch$; $Cost = Evaluate\ (ch')$;
Step3: while $(Temp > Final\ Temp)$do
$ch1 = Mutate\ (ch')$;
$NewCost = Evaluate\ (ch1)$;
$\Delta Cost = NewCost - Cost$;
if $(\Delta\ Cost \le 0)\ OR\ \left(e^{-\frac{\Delta Cost}{Temp}} > Rand\right)$ then
$Cost = NewCost$;
ch' = ch1;
endif
$Temp = cooling\ rate \times Temp$
endwhile
Step 4: Output: the best ch'.

Bees Algorithm (BA)

Ant colonies, beehives, bird flocks, and animal herds are examples of real-world swarm intelligence (SI). The three most popular examples of swarm intelligence systems are marriage in honey bee's optimization (MBO), particle swarm optimization, and ant colony optimization.

A novel technique known as MBO, which is applied to a particular class of propositional satisfiability issues, and it is based on the haploid-diploid genetic breeding of honey bees. The three primary MBO processes are: the queen bee's flight to mate with drones, her production of new broods, and her enhancement of the fitness of the broods. The difficulty lies in modifying the colony's self-organization behavior to address the issues. The Bees Algorithm (BA), a solution-finding algorithm, draws inspiration from honey bees' normal foraging behavior. In its most basic form, the pseudo code for the BA is as follows [11]:

The algorithm requires a number of parameters to be set, namely:

m: Number of scout bees.
ss: Number of sites selected out of n visited sites.
e: Number of best sites out of ss selected sites.
nep: Number of bees recruited for best e sites.
nsp: Number of bees recruited for the other (ss-e) selected sites.
ngh: Initial size of patches which includes site and its neighborhood and stopping criterion.

Algorithm (2): Bees Algorithm (BA)

Step 1: INPUT: m, ss, e, nep, nsp, Maximum of iterations.
Step 2: Initialize population with random solutions.
Step 3: Evaluate fitness of the population.
Step 4: REPEAT
Step 5: Select sites for neighborhood search.
Step 6: Recruit bees for selected sites (more bees for best e sites) and evaluate fitness's.
Step 7: Select the fittest bee from each patch.
Step 8: Assign remaining bees to search randomly and evaluate their fitness.
Step 9: UNTIL stopping criterion is met.
Step 10: OUTPUT: Optimal or near optimal solutions.
END.

Using Lsms for Solving the Two Bcmsp and Bomsp

In this Section, we suggest using LSMs. These LSM are Bees Algorithm (BA) and a Simulated Annealing (SA) to find the most efficient solutions for solving MCTSP. The values of $p_j\ and\ d_j$ for all examples are generated randomly s.t. $p_j \in [1,10]$ and $d_j \in \{[1,30], \quad 1 \le n \le 29. \ [1,40], \quad 30 \le n \le 99. \ [1,50], \ 100 \le n \le 999. [1,70], \quad n \ge 100.$
with condition $d_j \ge p_j$, for $j = 1,2, \dots, n$.

Now we introduce the following important abbreviations:

Ex: Example Number.
Av: Average.
AAE: Average Absolute Error.
$\frac{T}{S}$: Average of Time per second.
Av: Average.
R: $0 < Real\ < 1$.
F: Objective Function value for ER problem.

ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 34, Issue 4, 2023 DOI: https://doi.org/10.23851/mjs.v34i4.1430

F_1: Objective Function value for EPR problem.
G: Objective Function value for TR problem.
G_1: Objective Function value for TPR problem.
ES: efficient solution.
OS: optimal solution.

Applying LSM's for Solving the Two Problems
In this Section, we demonstrate the results of applying the two suggested algorithms (SA and BA) on the two problems (ER and TR).

Applying LSM's for Solving Problem (ER)
Before showing the results of applying the two LSM's for problem (ER), it is important to mention that the MST rule will be applied as:

1. Starting solution for SA.
2. One solution is the population of BA.

Table 1 shows the comparison results between CEM(ER, EPR) with SA(ER,EPR) and BA(ER,EPR) for $n = 3:2:11$. Table 2 shows the comparison results between BAB(ER, EPR) with SA(ER,EPR) and BA(ER,EPR) for $n = 30:30:300$. Table 3 shows the comparison results between DR(ER, EPR) and SA(ER,EPR) and BA(ER,EPR) for $n = 20,50,100,300,500,1000,2000,3000,4000$. Table 4 shows the comparison results between SA(ER, EPR) with BA (ER,EPR) for $n = 5000:1000:8000$.

Table 1. Comparison results between BAB(ER,EPR) with SA(ER,EPR) and BA(ER,EPR) for $n = 3:2:11$.

n	CEM(*ER*,*EPR*)			SA(*ER*,*EPR*)				BA(*ER*,*EPR*)			
	F	F_1	$\frac{T}{S}$	F	F_1	$\frac{T}{S}$	*AAE*	F	F_1	$\frac{T}{S}$	*AAE*
3	(7.0,5.6)	12.6	R	(7.0,5.6)	12.6	R	0	(7.0,5.6)	12.6	R	0
5	(5.8,14.2)	20	R	(5.8,14.2)	20.0	R	0	(5.8,14.2)	20.0	R	0
7	(6.4,18.2)	24.6	R	(6.4,18.2)	24.6	R	0	(6.4,18.2)	24.6	R	0
9	(3.4,20.0)	23.4	R	(3.4,20.0)	23.4	R	0	(3.4,20.0)	23.4	R	0
11	(6.0,37.4)	43.4	R	(6.0,37.4)	43.4	R	0	(6.0,37.4)	43.4	R	0
Av	(5.8,18.9)	24,8	R	(5.8,18.9)	24.8	R	0	(5.8,18.9)	24,8	R	0

Table 2. Comparison results between BAB(ER,EPR) and SA(ER,EPR) and BA(ER,EPR) for $n = 30:30:300$.

n	BAB(*ER*,*EPR*)			SA(*ER*,*EPR*)				BA(*ER*,*EPR*)			
	F	F_1	$\frac{T}{S}$	F	F_1	$\frac{T}{S}$	*AAE*	F	F_1	$\frac{T}{S}$	*AAE*
30	(2.8,127.8)	130.6	R	(2.8,127.8)	130.6	R	0	(3.0,128.4)	128.8	6.5	1.8
60	(1.0,294.8)	295.8	R	(1.0,295.0)	296.0	R	0.2	(1.0,295.6)	296.6	11.3	0.8
90	(1.2,469.6)	470.8	1.6	(1.2,469.8)	471.0	R	0.2	(1.2,470.0)	471.2	12.1	0.4
120	(0.6,622.6)	623.2	3.4	(0.6,622.8)	623.4	R	0.2	(0.6,622.8)	623.4	15.4	0.2
150	(1.4,786.4)	787.8	12.1	(1.4,786.8)	788.2	R	0.4	(1.5,787.7)	788.0	19.9	0.2
180	(0.6,905.2)	905.8	9.5	(0.6,905.4)	906.0	R	0.2	(0.6,905.4)	906.0	22.9	0.2
210	(1.0,1094.0)	1095	14.7	(1.0,1094.0)	1095.0	R	0	(1.0,1094.0)	1095.0	19.5	0
240	(0.6,1298.6)	1299.2	38.1	(0.6,1298.8)	1299.4	R	0.2	(0.6,1299.0)	1299.6	53.3	0.4
270	(0.4,1454.4)	1454.8	33.8	(0.4,1454.8)	1455.2	R	0.4	(0.4,1455.0)	1455.4	31.4	0.6
300	(1.0,1620.4)	1621.4	41.5	(1.0,1620.4)	1621.4	R	0	(1.0,1620.4)	1621.4	29.8	0
Av	**(1.1,705.4)**	**868.4**	**15.5**	**(1.1,867.6)**	**868.7**	**R**	**0.2**	**(1.1,867.8)**	**868.9**	**22.2**	**0.5**

Table 3. Comparison results between DR(ER,EPR) and SA(ER,EPR) and BA(ER,EPR) for $n = 20,50,100,300,500,1000,2000,3000,4000$.

n	MST-SPT-ERL			SA(*ER*,*EPR*)				BA(*ER*,*EPR*)			
	F	F_1	$\frac{T}{S}$	F	F_1	$\frac{T}{S}$	*AAE*	F	F_1	$\frac{T}{S}$	*AAE*
20	(2.8,90.6)	93.4	R	(3.0,89.8)	92.8	R	0.6	(2.8,89.2)	92.0	R	1.4
50	(2,235.4)	237.4	R	(2.0,234.0)	236.0	R	1.4	(2.2,235.0)	237.2	11.5	2.6

100	(0.8,505.6)	506.4	R	(0.8,505.2)	506.0	R	0.4	(0.8,505.6)	506.4	22.2	0
300	(1,1620.4)	1621.4	2.2	(1.0,1620.4)	1621.4	R	0	(1.0,1620.4)	1621.4	30.2	0
500	(0.2,2680.6)	2680.8	14.5	(0.2,2679.8)	2680.0	R	0.8	(0.2,2679.8)	2680.0	43.7	0.8
1000	(0.2,5435.6)	5435.8	15.5	(0.2,5435.6)	5435.8	R	0	(0.2,5435.6)	5435.8	105.5	0
2000	(0,10969.8)	10969.8	122.8	(0.0,10969.8)	10969.8	1.3	0	(0.0,10969.8)	10969.8	142.5	0
3000	(0.0,16447.6)	16598.0	61.9	(0.0,16447.6)	16447.6	1.7	150.4	(0.0,16447.6)	16447.6	133.3	150.4
4000	(0.0,21909.4)	21862.0	120.3	(0.0,21909.4)	21909.4	2.4	47.4	(0.0,21909.4)	21909.4	433.3	47.4
Av	(0.8,6655)	6667.2	37.5	(0.8,6654.6)	6655.4	0.6	22.3	(0.8,6654.7)	6655.5	102.5	22.5

Table 4. Comparison results between SA(ER,EPR) with BA (ER,EPR) for $n = 5000{:}\,1000{:}\,8000$.

n	SA (ER,EPR)			BA (ER,EPR)		
	F	F_1	$\frac{T}{S}$	F	F_1	$\frac{T}{S}$
5000	(0.0,27395.8)	27395.8	3.0	(0.0,27395.8)	27395.8	516.1
6000	(0.0,32901.8)	32901.8	3.4	(0.0,32901.8)	32901.8	502.3
7000	(0.0,38468.4)	38468.4	3.9	(0.0,38468.4)	38468.4	353.7
8000	(0.0,44006.0)	44006.0	4.4	(0.0,44006.0)	44006.0	616.8
Av	(0,35693)	35693	3.7	(0,35693)	35693	497.2

Applying LSM's for Solving Problem (TR)

Before showing the results of applying the two LSM's for problem (TR), it important to mention that the EDD rule will be applied as:

1. Starting solution for SA.
2. One solution is the population of BA.

Table 5 shows the comparison results between BAB(TR,TPR) with SA(TR,TPR) and BA(TR,TPR) for $n = 3{:}\,2{:}\,11$. Table 6 shows the comparison results between BAB(TR,TPR) and SA(TR,TPR) and BA(TR,TPR) for $n = 30{:}\,30{:}\,300$. Table 7 shows the comparison results between DR(TR,TPR) and SA(TR,TPR) and BA(TR,TPR) for $n = 20,50,100,300,1000,2000,3000,4000$. Table 8 shows the comparison results between SA(TR, TPR) with BA(TR,TPR) for

Table 5. The comparison results between BAB (TR,TPR) with SA (TR,TPR) and BA (TR,TPR) for $n = 3{:}\,2{:}\,11$.

n	CEM(TR,TPR)			SA(TR,TPR)				BA(TR,TPR)			
	G	G_1	$\frac{T}{S}$	G	G_1	$\frac{T}{S}$	AAE	G	G_1	$\frac{T}{S}$	AAE
3	(0.4,5.6)	6	R	(0.4,5.6)	6.0	R	0	(0.4,5.6)	6.0	R	0
5	(8.4,14.2)	22.6	R	(8.4,14.2)	22.6	R	0	(8.4,14.2)	22.6	R	0
7	(11.8,18.2)	30	R	(11.8,18.2)	30.0	R	0	(11.8,18.2)	30.0	R	0
9	(16.6,20.0)	36.6	R	(16.6,20.0)	36.6	R	0	(16.6,20.0)	36.6	R	0.2
11	(31.4,37.4)	68.8	R	(31.4,37.4)	68.8	R	0	(31.4,37.4)	68.8	1.4	0.2
Av	**(13.7,19.1)**	**32.8**	**R**	**(13.7,19.1)**	**32.8**	**R**	**0**	**(13.7,19.1)**	**32.8**	**0.3**	**0.1**

Table 6. Comparison results between BAB (TR,TPR) and SA(TR,TPR) and BA(TR,TPR) for $n = 30{:}\,30{:}\,300$.

n	BAB(TR,TPR)			SA(TR,TPR)				BA(TR,TPR)			
	G	G_1	$\frac{T}{S}$	G	G_1	$\frac{T}{S}$	AAE	G	G_1	$\frac{T}{S}$	AAE
30	(125.0,127.8)	252.8	R	(125.0,127.8)	252.8	R	0	(125.1,128.1)	245.6	7.4	7.2
60	(293.8,294.8)	588.6	R	(294.1,295.6)	589.7	R	1.1	(294.2,296.9)	591.1	11.3	2.5
90	(468.4,469.6)	938	1.47	(468.4,470.0)	938.4	R	0.4	(468.6,471.2)	944.3	13.2	6.3
120	(622.0,622.6)	1244.6	3.21	(622.0,622.6)	1244.6	R	0	(622.0,622.6)	1244.6	19.1	0
150	(785.0,786.4)	1571.4	5.5	(785.0,786.4)	1571.4	R	0	(785.0,788.2)	1573.2	22.6	1.8

180	(904.6,905.2)	1809.8	16.8	(904.6,905.8)	1810.4	R	0.6	(904.6,906.8)	1811.4	20.0	1.6
210	(1093.0,1094.0)	2187	14.1	(1093.0,1094.0)	2187.0	R	0	(1093.0,1094.2)	2187.2	16.9	0.2
240	(1298.0,1298.6)	2596.6	21.7	(1298.0,1299.0)	2597.0	R	0.4	(1298.0,1299.0)	2597.0	39.7	0.4
270	(1454.0,1454.4)	2908.4	29.0	(1454.0,1454.4)	2908.4	R	0	(1454.0,1454.4)	2908.4	55.9	0
300	(1619.4,1620.4)	3239.8	35.8	(1619.4,1620.8)	3240.2	R	0.4	(1619.4,1621.0)	3240.4	39.8	0.6
Av	**(866.3,867.4)**	**1733.7**	**12.8**	**(866.3,867.6)**	**1734.0**	**R**	**0.3**	**(866.4,868.2)**	**1734.3**	**24.6**	**2.1**

Table 7. Comparison results between DR (TR,TPR) and SA(TR,TPR) and BA(TR,TPR) for $n = 20,50,100,300,1000,2000,3000,4000$.

$\boldsymbol{n}$	**EDD-SPT-TRL**			**SA(*TR*,*TPR*)**				**BA(*TR*,*TPR*)**			
	$\boldsymbol{G}$	G_1	$\frac{T}{S}$	$\boldsymbol{G}$	G_1	$\frac{T}{S}$	$\boldsymbol{AAE}$	$\boldsymbol{G}$	G_1	$\frac{T}{S}$	$\boldsymbol{AAE}$
20	(86.4,89.2)	175.6	R	(86.4,89.2)	175.6	R	0	(86.4,89.2)	175.6	1.3	0
50	(231.6,233.6)	465.2	R	(231.6,233.8)	465.4	R	0.2	(231.6,235.0)	466.6	10.9	1.4
100	(504.4,505.2)	1009.6	R	(504.5,506.6)	1011.1	R	1.5	(504.4,506.6)	1011.0	29.1	1.4
300	(1619.4,1620.4)	3239.8	1.1	(1619.4,1620.8)	3240.2	R	0.4	(1619.4,1621.0)	3240.4	49.7	0.6
500	(2679.6,2679.8)	5359.4	2.5	(2679.6,2679.8)	5359.4	R	0	(2679.6,2679.8)	5359.4	64.1	0
1000	(5435.4,5435.6)	10871	13.2	(5435.4,5435.6)	10871.0	R	0	(5435.4,5435.6)	10871.0	76.4	0
2000	(10969.8,10969.8)	21939.6	73.6	(10969.8,10969.8)	21939.6	1.2	0	(10969.8,10969.8)	21939.6	211.5	0
3000	(16447.6,16447.6)	32895.2	40.4	(16447.6,16447.6)	32895.2	1.7	0	(16447.6,16447.6)	32895.2	256.8	0
4000	(21909.4,21909.4)	43818.8	69.6	(21909.4,21909.4)	43818.8	2.0	0	(21909.4,21909.4)	43818.8	361.5	0
Av	**(6653.7,6654.5)**	**13308.2**	**22.3**	**(6653.7,6654.7)**	**13308.4**	**0.6**	**0.2**	**(6653.7,6654.9)**	**13308.6**	**117.9**	**0.4**

Table 8. Comparison results between SA (TR, TPR) with BA(TR,TPR) for $n = 5000{:}\,1000{:}\,8000$.

$\boldsymbol{n}$	**SA(*TR*,*TPR*)**			**BA(*TR*,*TPR*)**		
	$\boldsymbol{G}$	G_1	$\frac{T}{S}$	$\boldsymbol{G}$	G_1	$\frac{T}{S}$
5000	(27395.8,27395.8)	54791.6	3.0	(27395.8,27395.8)	54791.6	147.2
6000	(32901.8,32901.8)	65803.6	3.4	(32901.8,32901.8)	65803.6	133.0
7000	(38468.4,38468.4)	76936.8	3.8	(38468.4,38468.4)	76936.8	173.9
8000	(44006.0,44006.0)	88012.0	4.3	(44006.0,44006.0)	88012.0	221.7
Av	**(35693,35693)**	**71386**	**3.6**	**(35693,35693)**	**71386**	**169.0**

RESULTS AND DISCUSSION

Problems (ER) and (EPR)

1. According to the results, the SA is more precise in results than the BA when compared to the BAB method for $30 \le n \le 300$ (see Table 2).
2. The SA is better than MST-SPT-ERL and BA in terms of time and good results for $20 \le n \le 4000$ (see table 3).
3. From Table 4, we see that the SA and BA outcomes are similar, but the SA performs better in terms of time for all n.

Problems (TR) and (TPR)

1. According to the results, the SA method is more precise than the BA method when compared to the BAB method for $30 \le n \le 300$ (see table 6).
2. The SA method is superior to the approximation method EDD-SPT-TRL in terms of time and good results for $20 \le n \le 4000$ (see table 7).
3. From table 8, we can see that the SA and BA methods produce outcomes that are similar, but the SA method performs better in terms of time.

CONCLUSIONS

In this research, we addressed the resolution of two classes of Multiple Constraints Multiple Sequence Problems (MCMSP), namely ER and TR, alongside two instances of Multiple

Objective Multiple Sequence Problems (MOMSP), denoted as EPR and ETR, employing Latent Semantic Models (LSMs). Our investigation underscored the commendable efficacy of two specific LSMs, denoted as SA and BA, in resolving both MCMSP and MOMSP. Comparative analyses against established benchmarks such as CEM, BAB, and various heuristic methods revealed the superior performance of our proposed LSMs. Throughout our experimentation, we observed the noteworthy influence of initial solutions on the attainment of optimal outcomes for SA and BA across varying parameters denoted as 'n', a trend evident across all result tables. Notably, SA exhibited heightened precision in results and consumed lesser CPU time in comparison to BA and other methodologies under evaluation. To further enhance the performance and efficacy of Latent Semantic Models, we advocate for the development of a hybrid approach integrating the strengths of both SA and BA, specifically tailored to address the discussed problem sets. Additionally, we propose the integration and exploration of additional LSMs, such as Tabu Search and Particle Swarm Optimization, to tackle and resolve the complexities inherent in the studied case problems.

ACKNOWLEDGMENTS

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

Disclosure and Conflict of Interest: The authors declare that they have no conflicts of interest.

REFERENCES

[1] H. J. Mutashar, Flow Shop Scheduling Problems Using Exact and Local Search Methods, Baghdad: M.Sc. Thesis, Dept. of Mathematics, College of Science, Mustansiriyah University, 2008.

[2] S. J. Ibraheem, Single Machine Multicriteria Scheduling, Baghdad: M. Sc. Thesis, Mathematics department, College of Ibn AL-Haitham, University of Baghdad, 2014.

[3] A. A. Mahmood, "Local Search Algorithms for Multiobjective Scheduling Problem," Journal of Al Rafidain University College, no. 36, 2015.

[4] S. B. Abdulkareem, Algorithms for Solving Multi-Criteria Scheduling Problems, Baghdad: Mathematics department, College of Science, Mustansiriyah University, 2017.

[5] D. A. Abbass, "Using Branch and Bound and Local Search Methods to Solve Muiti-Objective Machine Scheduling Problem," in First International Conference of Computer and Applied Sciences (CAS), IEEE Xplore, 2019. https://doi.org/10.1109/CAS47993.2019.9075460

[6] A. M. Gh, Exact and Approximation Methods for Solving Combinatorial Optimization Problems, Baghdad, Iraq.: Ph. D. Thesis, Mathematics department, College of Science, Mustansiriyah University, 2022.

[7] G. A. Manal and F. H. Ali, "Optimal and Near Optimal Solutions for Multi Objective Function on a Single Machine," in 1st International Conference on Computer Science and Software Engineering (CSASE2020), Sponsored by IEEE Iraq Section, 16-17 Apr., Duhok, Kurdistan Region - Iraq, 2020.

[8] S. F. Yousif and F. H. Ali, "Solving Maximum Early Jobs Time and Range of Lateness Jobs Times Problem Using Exact and Heuristic Methods," Iraqi Journal of Science, (accepted 2023), vol. 65, no. 2, 2024.

[9] M. H. Ibrahim, Exact and Approximation Methods for Solving Machine Scheduling Problems with and without Setup Time, Baghdad: Ph. D. Thesis, Mathematics department, College of Science, Mustansiriyah University, 2022.

[10] S. M. Jasim and F. H. Ali, "Exact and Local Search Methods for Solving Travelling Salesman Problem with Practical Application," Iraqi Journal of Science, [S.l], vol. 60, no. 5, pp. 1138-1153, 2019. https://doi.org/10.24996/ijs.2019.60.5.22

[11] F. A. Ali, Improving Exact and Local Search Algorithms for Solving Some Combinatorial Optimization Problems, Baghdad: Ph. D. Thesis, Dept of Mathematics, College of Science, Al-Mustansiriyah University, 2015.

How to Cite

S. F. . Yousif, F. H. Ali, and K. F. . Alshaikhli, "Using Local Search Methods for Solving Two Multi-Criteria Machine Scheduling Problems", *Al-Mustansiriyah Journal of Science*, vol. 34, no. 4, pp. 96–103, Dec. 2023.