DOI: 10.24996/ijs.2023.64.7.29

ISSN: 0067-2904

Exact and Near Pareto Optimal Solutions for Total Completion Time and Total Late Work Problem

Faez Hassan Ali[1], **Rasha Jalal Mitlif**[2], **Wadhah Abdulleh Hussein**[3]
[1]*Mustansiryah University, College of Science, Mathematics Dept., Baghdad, Iraq*
[2]*Branch of Mathematics and Computer Applications, Department of Applied Sciences, University of Technology, Baghdad, Iraq*
[3]Mathematics department, College of Science, University of Diyala, Iraq

Received: 1/8/2022 Accepted: 26/9/2022 Published: 30/7/2023

Abstract
In this paper, the bi-criteria machine scheduling problems (BMSP) are solved, where the discussed problem is represented by the sum of completion and the sum of late work times $(1//(\sum C_j, \sum V_j))$ simultaneously. In order to solve the suggested BMSP, some metaheurisitc methods are suggested which produce good results. The suggested local search methods are simulated annulling and bees algorithm. The results of the new metaheurisitc methods are compared with the complete enumeration method, which is considered an exact method, then compared results of the heuristics with each other to obtain the most efficient method.

Keywords: Bi-criteria machine scheduling problems, Complete enumeration method, completion time, Late work times.

حلول باريتو الدقيقة والتقريبية لمسألة مجموع اوقات الاتمام ومجموع اوقات الاعمال المتاخرة

فائز حسن علي[1]، رشا جلال متلف[2]، وضاح عبد اللة حسين[3]
[1]الجامعة المستنصرية، كلية العلوم، قسم الرياضيات، كلية العلوم، الجامعة المستنصرية، بغداد، العراق.
[2]قسم الرياضيات وتطبيقات الحاسوب , قسم العلوم التطبيقية, الجامعة التكنولوجية , بغداد، العراق.
[3]قسم الرياضيات, كلية العلوم، جامعة ديالى, العراق

الخلاصه
في هذا البحث تم ايجاد بعض الطرق لحل واحدة من مسائل جدولة الماكنة ثنائية المعايير (BMSP). ان المسألة المراد مناقشتها تتمثل بمسألة مجموع اوقات الاتمام ومجموع الاعمال المتاخرة انيا $(1//(\sum C_j, \sum V_j))$. لحل هذه المسألة, تم اقتراح فوق الحدسية والتي اعطت نتائج جيدة. ان الطرق المقترحة لحل المسالة المذكورة هي طريقة محاكاة التلدين وخوارزمية النحل. ان نتائج الطرق الحدسية المقترحة تم مقارنتها مع طريقة العد التام ومن ثم تم مقارنة نتائج تلك الطرق المقترحة مع بعضها لتحديد اي الطريقتين هي الاكفأ.

1. Introduction
Machine scheduling optimization problems with two criteria are based on competing for objective functions, a set is formed and called the Efficient Pareto optimal solutions set, which is regarded as a vicarious of one optimal solution. This set contains one (or more) solution(s)

*Email: faezhassan@uomustansiriyah.edu.iq

that, according to the objective functions, are superior to any other solution(s). In the literature, there are two approaches for multicriteria scheduling problems [1]; the simultaneous approach and the hierarchical approach.

The most important in the last five years' literature surveys are: some efficient algorithms are suggested for solving the BMSP. Ali and Abdul-Kareem (2017) [2], in their paper, some kinds of local search methods (LSM): Bees algorithm (BA) and particle swarm optimization (PSO) are used to minimize $(T_{max}, V_{max}, \sum V_j)$ simultaneously.

Gallo and Capozzi (2019) [3] used Simulated Annealing (SA) to solve MSP on m machines to minimize the total completion time $\left(\sum\sum C_j^k\right)$. SA and its key parameters (tempering, freezing, cooling, and the number of contours to be explored) are investigated, and the choices made in identifying these parameters are illustrated in order to generate a good algorithm that efficiently solves the MSP.

Ali and Ahmed (2020) [4] introduced a multicriteria objectives function $1//\left(\sum C_j, R_L, T_{max}\right)$ P-problem in a single MSP which is solved by BAB and some heuristic methods. Some special cases are introduced and proved to find efficient solutions to problems. Then they solved $1//\left(\sum C_j + R_L + T_{max}\right)$ P_1-problem to find optimal or good solutions by using exact and heuristic methods [5]. Lastly, the BA and PSO are used for solving the two suggested problems [6].

Ali et. al. (2021) [7] implemented the Neural Networks (NN) to manipulate the MSP $1//(T_{max}, V_{max}, \sum V_j)$ simultaneously. The results prove the efficiency of NN which is learned by back propagation algorithm for $n \leq 500$ jobs.

Ibrahim et. al. (2022) [8] proposed a BAB method to solve the multi-objective function (MOF) problem $1//\left(\sum E_j + T_j + C_j + U_j + V_j\right)$. Also, they use fast LSMs (SA and Genetic Algorithms (GA)) yielding near optimal solution. they report on computation experience; the performances of the exact and LSMs are tested on a large class of test problems.

The mathematical formulation of $1//\left(\sum C_j, \sum V_j\right)$ is discussed in section two we will discuss problem (CV-problem) and its subproblem $\left(1//\sum\left(C_j + V_j\right)\right)$ problem (CV_1-problem). The simulated annealing and Bees Algorithm are introduced as a metaheuristic method to solve the two problems which are introduced in section three. Section four introduces the comparative and the practical and results. In section five, we will present the discussion of the practical results. Finally, in section six, some conclusions and recommendations are presented. In this manner we introduce 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, where $C_j = \sum_{k=1}^{j} p_k$.
T_j	:	The tardiness of job j, $T_j = \max\{C_j - d_j, 0\}$.
V_j	:	The late work of job j, where $V_j = min\{p_j, T_j\}$.
$\sum C_j$	:	Total completion time.
$\sum V_j$	:	Total late work.
OP	:	Optimal Value of CV_1-problem.

Definition (1) [9]: A schedule S is said to be efficient if another schedule S' cannot satisfy $f_j(S') \leq f_j(S),\ j = 1, \dots, k$, with at least one of the above holding as a strict inequality. Another way to say this is that S' dominates S.

Definition (2) [10]: In a multicriteria resolution, the term "optimize" refers to a solution in which there is no way to develop or improve one objective without worsening the other.
Definition (3): Let (f_0, g_0) be a solution for multi-criteria problem $1//(f, g)$, then the Euclidean distance (d) for this solution is:

$$d = \sqrt{f_0^2 + g_0^2} \qquad \dots(1)$$

Remark (1): The d distance can be a good measure to find the best efficient solution from the set of Pareto optimal set.

Proposition (1): Let (f_0, g_0) be a solution for multi-criteria problem $1//(f, g)$, and $f_0 \neq 0$ and $g_0 \neq 0$ then always: $f_0 \leq d$.
Proof: Let's assume that $f_0 > d$, from (1):

$$f_0 > \sqrt{f_0^2 + g_0^2}$$

$$f_0^2 > f_0^2 + g_0^2$$

This is a contradiction since the above inequality is not true even $f_0 = 0$ and $g_0 = 0$.

2. The $1//(\sum C_j, \sum V_j)$ Problem with Mathematical Formulation

Consider we have a single machine, with set $N = \{1,2, \dots, n\}$ with n jobs, let $\sigma \in S$ which is the set of all feasible schedules. We want to minimize the problem $(\sum C_j, \sum V_j)$, which is formulated as follows:

$$\left.\begin{array}{ll} Min\{\sum C_j, \sum V_j\} & \\ \text{Subject to,} & \\ C_j \geq p_{\sigma(j)}, & j = 1,2, \dots, n. \\ C_j = C_{(j-1)} + p_{\sigma(j)}, & j = 2,3, \dots, n. \\ T_j \geq C_j - d_{\sigma(j)}, & j = 1,2, \dots, n. \\ V_j = min\{p_j, T_j\}, & j = 1,2, \dots, n. \\ T_j, V_j \geq 0, & j = 1,2, \dots, n. \end{array}\right\} \qquad \dots(CV)$$

For CV-problem, we can deduce subproblem: The $1//\sum C_j + \sum V_j$ Problem:

$$\left.\begin{array}{ll} Min\{\sum C_j + \sum V_j\} & \\ \text{Subject to,} & \\ C_j \geq p_{\sigma(j)}, & j = 1,2, \dots, n. \\ C_j = C_{(j-1)} + p_{\sigma(j)}, & j = 2,3, \dots, n. \\ T_j \geq C_j - d_{\sigma(j)}, & j = 1,2, \dots, n. \\ V_j = min\{p_j, T_j\}, & j = 1,2, \dots, n. \\ T_j, V_j \geq 0, & j = 1,2, \dots, n. \end{array}\right\} \qquad \dots(CV_1)$$

The problems $CV\ and\ CV_1$ are NP-hard because of $\sum V_j$ is NP-hard.

Example 1:
From Remark (1), we check the usefulness of the d distance by using the following scheduling data:

	1	2	3	4
p_i	10	5	9	2
d_i	13	28	24	29

After applying the Complete Enumeration Method (CEM) for this data, we obtain (4) efficient solutions as shown in Table 1.

Table 1: Efficient solutions for example (1) with distance d.

i	Efficient Sequence	Efficient Solution	d
1	4,2,3,1	(51,10)*	51.97
2	4,2,1,3	(52,6)#	52.35
3	4,1,2,3	(57,2)	57.04
4	4,1,3,2	(61,0)	61.00

Notice that, for CV-problem, the first efficient solution has the best d among all efficient solutions (signed with *), while for CV_1-problem, the optimal solution is the second one (signed with #).

3. Metaheuristic Methods

In this section, we will discuss two metaheuristic methods to solve CV and CV_1; these two methods are simulated annealing and the Bees Algorithm.

3.1 Simulated Annealing

The physical annealing process is represented by simulated annealing (SA) [3]. This name refers to the simulation of the annealing process, which is associated with a temperature-decreasing annealing schedule. SA is a local optimization technique in which the initial solution is always improved by small local effects until none of these effects can improve the solution any further.

The initial state or solution of a thermodynamic system was chosen at energy ($Cost$) and temperature as the original Metropolis acceptance criterion ($Temp$ or t). Keeping constant t, the initial setting of the system is perturbed to produce a new setting and the energy ΔC is calculated. If ΔC is negative, the new setting is accepted without conditions; otherwise, it is accepted with a probability determined by the Boltzmann factor $e^{-\Delta C/Temp}$ to stay away from trapping in the local optima. A simple scheme of SA [11] is as follows:

<u>Simulated Annealing Algorithm</u>

$[ch']$=SA(ch)
    $ch' = ch$;
    $Cost$ = Evaluate(ch');
    $Temp = InitialTemperature$;
    WHILE ($Temp > FinalTemperature$)
        ch_1 = Mutate(ch');
        $NewCost$ = Evaluate(ch_1);
        $\Delta C = NewCost - Cost$;
        IF ($\Delta C \leq 0$) OR ($e^{-\Delta C/Temp} > Rand$) THEN

$Cost = NewCost;$
$ch' = ch_1;$
ENDIF
$Temp = cooling\ rate \times Temp;$
END{WHILE}
Return the best solution;
END

It's important to mention that:

- $cooling\ rate$ is 0.95.
- $Temperature$ is 10000.
- $Temperature$ is 0.
- $Rand$ as a uniform real random.
- ch is the chromosome, in MSP its represent the sequence of scheduling, for instance, for $n = 5, ch = [3\ 4\ 5\ 2\ 1]$.

3.2 Bees Algorithm (BA)

The main processes in Bees Algorithm (BA) are the queen bee's mating flight with drones, the queen bee's creation of new broods, worker fitness improvement, worker adaptation, and the replacement of the least fit queen with the fittest brood [12].
The challenge is to adapt the colony's self-organization behavior to problem solving. The BA is an optimization algorithm inspired by honey bee foraging behavior to find the best solution. [13].
The most important parameters of BA are:

k	:	**Number of scout bees which are be selected randomly.**
m	:	Number of sites of flowers which are selected out of n visited sites.
e	:	Number of best sites which are selected out of m site randomly.
nep	:	Number of bees recruited for best e sites.
nsp	:	Number of bees recruited for the remaining $(m - e)$ selected sites.
ngh	:	Initial size of patches which includes site and its neighborhood and stopping conditions.

The main steps of BA are as follows:

Bees Algorithm
INPUT: k, m, e, nep, nsp, Maximum of iterations.
Step1. Initialization of random solutions population.
Step2. Evaluate fitness of each solution (individual) in the population.
Step3. WHILE stopping criterion is not met
Step4. Select sites for neighborhood search.
Step5. Choosing recruit bees for the selected sites and evaluate fitness's.
Step6. Select the fittest or best bee from each patch.
Step7. Assign remaining bees to search randomly and evaluate their fitness's.
Step8. END{WHILE}.
OUTPUT: Best solutions.
END.
Note: The random solutions (RS) in population of MSP is the random sequence of scheduling, for instance, for $n = 4, RS = [4\ 1\ 3\ 2]$.

The advantages of Bees algorithm [19]:

- BA is more scalable; it takes less time to complete the objective.

- BA is more efficient at finding and collecting food because it requires fewer steps.

4. Practical Results of *CV* and *CV*1-Problems

Since we deal with the MSP, so the p_j and d_j values are generated randomly for five examples s.t. $p_j \in [1,10]$ and:

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

with condition $d_j \ge p_j$, for $j = 1,2,\dots,n$.
Now, we introduce the following abbreviations:

Ex	:	**Example Number.**
Av	:	Average.
ANS	:	Average number of efficient solutions.
AAE	:	Average Absolute Error.
AT	:	Average of Time per second.
Ad	:	Average of Euclidean distance.
ASOF	:	Average Single Objective Function.
F_1	:	Objective Function value for CV_1-problem.
AMOF	:	Average Multi Objective Function.
F	:	Objective Function value for *CV*-problem.
R	:	$0 <$ Real < 1.

4.1 Comparison Results of *CV*-problem.

The CEM, BA and SA methods all were tested by programming them using MATLAB ver2017R.
Comparison efficient results between CEM(F) with LSM: BA(F) and SA(F) for *CV*-problem are shown in table (2), for $n = 4:11$.

Table 2: Comparison between CEM(F), BA(F) and SA(F) for *CV*-problem, $n = 4:11$.

n	CEM(F)				BA(F)				SA(F)			
	AMOF	***AT***	***ANS***	***Ad***	***AT***	***ANS***	***Ad***	***AAE***	***AT***	***ANS***	***Ad***	***AAE***
4	(54.9,7. 0)	*R*	2.6	55.5	*R*	2.6	59.7	(1.7,0.1)	*R*	2.6	55.5	(0,0)
5	(64.6,8.3)	*R*	2.2	65.3	*R*	3.4	72.5	(5,0.8)	*R*	2.0	65.0	(0.4,0.1)
6	(99.9,11.3)	*R*	5.0	100.7	1	2.8	114.8	(3.2,1.6)	*R*	4.0	101.0	(1.4,1.3)
7	(131.0,16.5)	*R*	5.4	132.2	1.0	2.8	155.0	(10,1.4)	R	4.0	131.6	(4.4,0.2)
8	(206.1,29.7)	*R*	3.2	208.2	1.1	2.8	243.6	(23.5,1.9)	*R*	3.0	211.6	(6.9,0.3)
9	(230.9,31.8)	3.7	3.6	233.1	1.1	3.4	258.5	(34.2,3.8)	*R*	2.4	241.0	(1,0.5)
10	(202.4,27.7)	41.2	5.6	204.3	1.1	3.4	261.6	(47.8,1.7)	*R*	3.0	230.3	(0.7,0.9)
11	(298.7,42.4)	470.6	3.8	301.8	1.2	2.8	362.9	(74.8,0.7)	*R*	3.4	307.7	(21.5,0.1)
Av	**(161.0,21.8)**	**64.4**	**3.9**	**162.6**	**0.8**	**3**	**191.1**	**(25.0,1.5)**	**0**	**3.1**	**168.0**	**(4.5,0.4)**

By using Table 2, Figure 1 shows the *Ad* values for results of CEM(F), BA(F) and SA(F), for *CV*-problem, for $n = 4:11$.

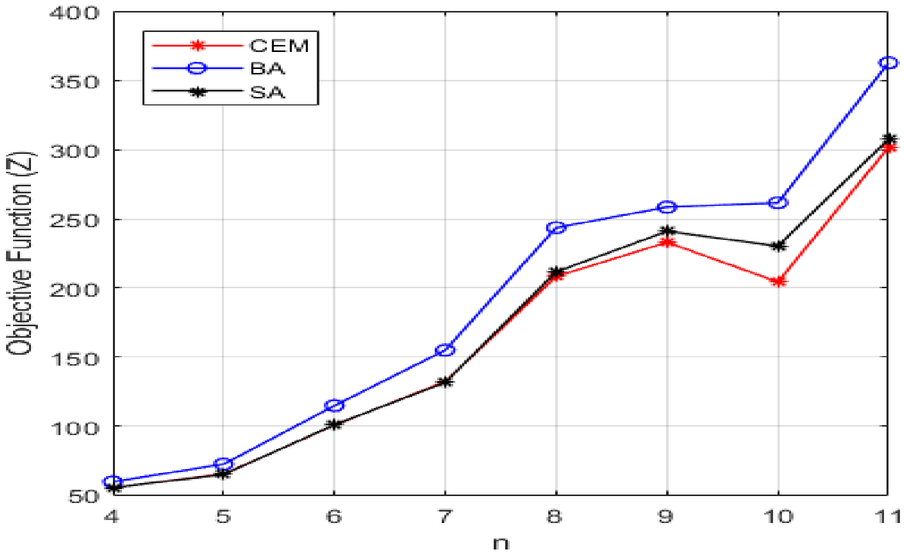

Figure 1: Comparison results of Ad for CEM(F), BA(F) and SA(F) for $n = 4{:}11$.

The comparison results between BA(F) and SA(F) for CV-problem for $n = 30,70,100,300,700,1000,3000$ are shown in Table 3.

Table 3: a comparison results between BA(F) and SA(F) for CV-problem for different n.

n	BA(F)				SA(F)			
	$AMOF$	AT	ANS	Ad	$AMOF$	AT	ANS	Ad
30	(2456.8,139.5)	2.0	3.4	2543.7	(1964.4,134.8)	R	5.4	1979.7
70	(12933.5,355.1)	5.1	3.4	13442.4	(9922.6,347.9)	1.5	3.4	10019.9
100	(27478.0,526.4)	6.8	2.6	27587.1	(21535.2,520.6)	1.9	7.2	20604.1
300	(244350.1,1615.5)	19.2	3.4	249521.8	(235625.2,1609.6)	5.2	4.6	234054.9
700	(1336548.6,3806.6)	35.9	4.2	1348243.8	(1333398.8,3800.0)	10.4	2.2	1324956.1
1000	(2721058.4,5445.0)	39.9	3.0	2759528.0	(2754414.6,5447.8)	14.4	2.4	2729210.4
3000	(24566261.5,16433.5)	103.8	3.0	24675496.5	(24720184.2,16433.5)	43.2	1.2	24678990.9

4.2 Comparison Results of CV_1-problem.

The optimal results of CEM(F_1) are compared with results of BA(F_1) and SA(F_1), $n = 4{:}11$, for CV_1-problem, these results are shown in Table 4.

Table 4: Comparison between CEM(F_1) and BA(F_1) and SA(F_1), $n = 4{:}11$, for CV_1-problem.

n	CEM(F_1)	BA(F_1)	SA(F_1)

	OP	***AT***	***ASOF***	***AT***	***AAE***	***ASOF***	***AT***	***AAE***
4	60.0	*R*	60.0	*R*	0	60.0	*R*	0
5	71.6	*R*	71.6	*R*	0	71.6	*R*	0
6	106.0	*R*	106.0	*R*	0	108.4	*R*	2.4
7	142.2	*R*	142.8	*R*	0.6	143.2	*R*	1
8	233.0	*R*	233.8	*R*	0.8	235.0	*R*	2
9	259.8	5.7	265.0	*R*	5.2	261.8	*R*	2
10	225.0	66.9	244.4	*R*	19.4	225.2	*R*	0.2
11	339.4	667.4	357.0	*R*	17.6	341.0	*R*	1.6
Av	**179.6**	**92.5**	**185.1**	*R*	**5.5**	**180.8**	*R*	**1.2**

Notice that the heuristics BA(F_1), and SA(F_1) give good objective values compared with CEM(F_1), and that can be noticed from AAE, for CV_1-problem.

For CV_1-problem, Figure 2 shows the comparison results of CEM(F_1), BA(F_1) and SA(F_1). All these results are obtained from Table 4, for number of jobs $n = 4{:}11$.

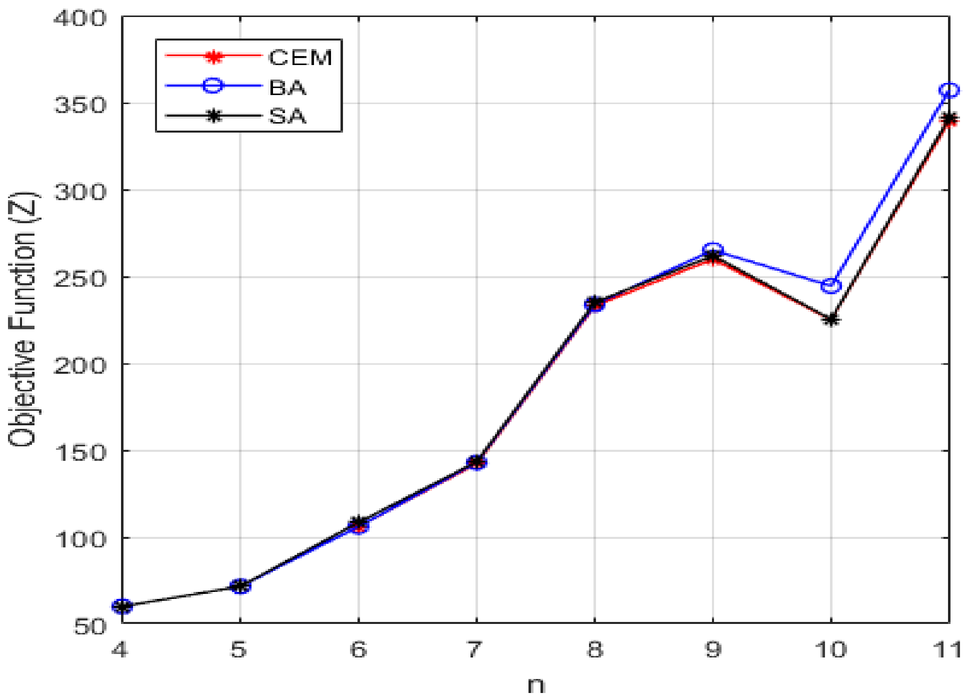

Figure 2: Comparison results of CEM(F_1), BA(F_1) and SA(F_1) for $n = 4{:}11$.

Table 5 describes the average of best solutions for CV_1-problem for $n =$ 30,70,100,300,700,1000 and 3000, using BA(F_1) and SA(F_1).

Table 5: Results of comparison of BA(F_1) and SA(F_1) for (CV_1), for different n.

n	BA(F_1)		SA(F_1)	
	ASOF	***AT***	***ASOF***	***AT***
30	2428.2	*R*	2019.8	*R*
70	12869.0	1.4	9935.4	*R*
100	27305.4	1.9	20476.4	*R*
300	241951.2	7.1	235602.4	2.0
700	1322347.0	5.0	1338330.4	2.8
1000	2697302.8	8.7	2714383.6	3.1
3000	24488108.6	14.6	24788317.6	7.5
Av	**4113187.0**	**5.5**	**4158438.3**	**2.2**

5. Evaluation of Practical Results of The Suggested Problems

1. For CV-problem:

a. The SA(F) is better than BA(F) in accuracy compared with CEM(F) and in CPU-time they approximate each other (see tables (2)) for all $n \leq 11$**.**

b. We notice that SA(F) has good accuracy compared with BA(F), for $30 \leq n \leq 700$, while BA(F) is better for $1000 \leq n \leq 3000$. In CPU-time SA(F) is better for all different n , see Table 3**.**

2. For CV_1-problem:

a. In accuracy, we see that SA(F_1) is relatively better than BA((F_1) in accuracy for $n \leq 11$ compared with CEM((F_1), and so on in CPU-time , see Table 4.

b. We see that SA(F) has better accuracy compared with BA(F), for $30 \leq n \leq 300$, while BA(F) is better for $700 \leq n \leq 3000$. In CPU-time SA(F) is better for all n (see tables (5))**.**

6. Conclusions and Future Work

1. The practical results of this paper show the efficiency of the two suggested methods: BA and SA for the two problems.
2. For CV-problem, $n \leq 700$, the performance of SA is better than BA in accuracy, while BA is better than SA for $n > 700$, and SA is better CPU-time for all n.
3. For CV_1-problem, $n \leq 300$, the performance of SA is better than BA in accuracy, while BA is better than SA for $n > 300$, and SA is better CPU-time for all n.
4. To increase the efficiency of the two LSMs, we suggest a hybrid between SA and BA to solve the two problems CV and CV_1.
5. For future work, we suggest using other local search methods (like Ant colony algorithm, genetic algorithm, particle swarm optimization…, etc) to find efficient and approximation solutions for CV and CV_1-problem for $n > 100$.

Acknowledgments

The authors would like to thanks Mustansiriyah university - Baghdad – Iraq for its support in the present work.

References:

[1] T. S. Abdul-Razaq and F. H. Ali, "Algorithm for Scheduling a Single Machine to Minimize Total Completion Time and Total Tardiness", *The 2nd International Conference on Mathematical Sciences-ICMA*, 2013, pp.23-24.

[2] F. H. Ali and Sh. B. Abdul-Kareem, "Local Search Methods to Solve Multi-Criteria Machine Scheduling Problems", *the 23th Scientific Conference, Al-Mustansiriyah University, College of Education*, 2017, pp.26-27.

[3] C. Gallo and V. Capozzi, , “A Simulated Annealing Algorithm for Scheduling Problems”, *Journal of Applied Mathematics and Physics*, vol.7, no.11, 2019.

[4] F. H. Ali and M. Gh. Ahmed, “Efficient Algorithms to Solve Tricriteria Machine Scheduling Problem”, (15th and the second International) Conference of Statistical Applications (ICSA2020), Irbil, Kurdistan Region–Iraq, *Journal of Al Rafidain University* College, Issue No. 46, 2020.

[5] F.H. Ali and M. Gh. 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*, 2020.

[6] F. H. Ali and M. Gh. 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*, 2020.

[7] F. H. Ali, H. C. Ali and Sh. B. Abdulkareem, “Neural Network for Minimizing Tricriteria Objective Function for Machine Scheduling Problem”, *Ibn Alhaitham 2nd International Conference for Pure and Applied Science (IHICPAS)-2020. Journal of Physics: Conference Series* 1879 (2021) 032023 IOP Publishing, 2021.

[8] M. H. Ibrahim, F. H. 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, vol.13 , no. 1, pp. 1649-1658, 2022.

[9] J. A. Hoogeveen, “Single Machine Scheduling to Minimize a Function of Two or Three Maximum Cost Criteria”, *Journal of Algorithms*, vol. 21, pp. 415-433, 1996.

[10] 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, 1991.

[11] S. M .JASIM, 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.

[12] A .Ashraf, P. Michael and C. Marco, “*Bees Algorithm”, Manufacturing Engineering Center, Cardiff University, Wales, UK*, 2009.

[13] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri and M. Zaidi, “The Bee's Algorithm – a Novel Tool for Complex Optimization Problems”, *2nd Virtual International Conference on Intelligence Production Machines and Systems (IPROMS 2006). Elsevier, Oxford*, 2006, pp. 454-459.