Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 1709-1720
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.5786

Search method for solving multicriteria scheduling problem

Doha Adel Abbas[a,*]

[a]*Department of Mathematics, College of Science, University of Al-Mustansiriyah, Bagdad, Iraq*

(Communicated by Madjid Eshaghi Gordji)

Abstract

Our research includes studying the case $1//F(\sum U_i, \sum Ti, T_{max})$ minimized the cost of a three-criteria objective function on a single machine for scheduling n jobs. and divided this into several partial problems and found simple algorithms to find the solutions to these partial problems and compare them with the optimal solutions. This research focused on one of these partial problems to find minimize a function of sum cost of $(\sum U_i)$ sum number of late job and $(\sum Ti)$ sum Tardiness and (T_{max}) the Maximum Tardiness for n job on the single machine, which is NP-hard problem, first found optimal solutions for it by two methods of Complete Enumeration technique(CEM) and Branch and Bounded ((BAB)). Then use some Local search methods(Descent technique(DM), Simulated Annealing (SA) and Genetic Algorithm (GA)), Develop algorithm called ((A)) to find a solution close to the optimal solution. Finally, compare these methods with each other.

Keywords: Descent Method(DM), Genetic Algorithm(GA), Maximum tardiness, Multi-objective optimization, Simulated annealing ((SA)), Total Number of Late job, Total Tardiness.

1. Introduction

For several years, researchers focused on single regular performance measurement on single machine scheduling problem has been widely during the Past decades. Until the late 1980, it was common practice the objective function one criterion was taken into account. If only one criterion is taken into account, then the result to be unbalanced, regardless of which standard is considered. If every work-in-process inventories low, then job are like to be completed far beyond their due dates. In order to reach an acceptable middle solution, then measure the goodness of a solution on all criteria important. That led to the development of scheduling multicriteria[13].

*Corresponding author
Email address: doha.a.a@uomustansiriyah.edu.iq (Doha Adel Abbas)

Received: March 2021 *Accepted:* May 2021

However, decision makers get up scheduling according to more than one criteria. Since us the multiple criteria is more real, then appeared several multicriteria scheduling articles in the scheduling problem. some of these reserch are bicriteria and find the minimization of two criteria. Smith (1956) studied of $1\ /\bar{d}/F(\sum Ci, T_{max})$ problem where $T_{max} = 0$ [6]. Vanwassenhove and Gelders extended this problem (1980) to the $1//F(\sum Ci, T_{max})$ with $T_{max} \neq 0$. The set of active points is describe and a pseudo-polynomial algorithm to list of all point is given. Nelson (1986) expanded a BAB algorithm to solve the problem for $(\sum Ci), (\sum U_i)$ and (T_{max}) at the same time. Hoogveen and Van de Velde (1995) showed that exactly the same approach can be used to solve the $1//F(\sum Ci, f_{max})$ problem. Tadie et at 2002, suggest that takes of an algorithm for finding the Pareto-optima set by apply specially develop constraints to a BAB algorithm for the $1//F(\sum Ti, T_{max})$ problem [12].To find the set of efficient point for $1//F(\sum Ci, E_{max})$ problem, [14] used genetic algorithm and [7] proposed a polynomial algorithm within special range. Recently, [19] have used a heuristic approach for minimizing total completion time and number of tardy jobs simultaneously on single machine with release date (i.e., for the $1/ri/F(\sum Ci, U_i)$ problem. [2] presented algorithms for many bicriteria scheduling problems on a single machine with release dates. The two criteria to be minimized are C_{max} and $\sum Ci$. He presented optimal solution for the two hierarchical problems of the $1/ri/F(\sum Ci, C_{max})$ problem. Recently multicriteria scheduling problems has been studied by several researchers in different directions [10, 15, 16, 5, 8].

In this paper, took the case of scheduling a collection $N = \{1, 2, \ldots, n\}$ on a single machine for the three criteria $1//F(\sum U_i, \sum Ti, T_{max})$ and divide them into partial problems and find algorithms to solve them and focus calculated the minimize value of the sum of the three criteria $\sum U_i, , \sum Ti$ and T_{max} (i.e $(\sum U_i + \sum Ti + T_{max}))$ on the single machine. related with job i its processing time pi and its due date di using methods to find the exact solution such as the (CEM) and (BAB) method and compared it with the approximate solution methods and calculate the time Ti it takes in each of the methods. The main object is to find a collection of near optimal solutions for the $1//(\sum U_i + \sum Ti + T_{max}))$ problem.

This paper begins with some basic scheduling concepts of multicriteria problems and basic rules are given in section 2. Formulations and some algorithms are given in section 3. (BAB) for the $1//(\sum U_i + \sum Ti + T_{max}))$ is given in section 4. Algorthm ((A)) in the section 5.Local search and genetic algorithm for the problem $1//(\sum U_i + \sum Ti + T_{max}))$ are given in sections 6. In section 7 computational experiments is given.

2. Basic scheduling concepts and basic rules [18]

The following notation will be used in this paper, jobs j $(j = 1, \ldots, n)$ then

- n : number of a jobs.
- P_j : This means the processing time.
- d_j : this means the du date.
- LB : lower bound
- UB : Upper bound
- BAB : branch and bound
- CEM : Complete Enumeration method

Now, compute for job j

C_j : this mean Earliest Completion time.
$T_j = max\{C_j - dj, 0\}$ The tardiness.
$U_j = \begin{cases} 0 & \text{if } C_j \leq d_j \\ 1 & \text{o.w} \end{cases}$ The unit penalty.
$T_{max} : Max\{Tj\}$, the maximum tardiness
$\sum T_j : sum\{Tj\}$, the total tardiness.

The simple rules for scheduling:

1. **SPT** order: It solve case $1//\sum Ci$ this Order of jobs in increasing of P_i for all job i.

2. The earliest due date **EDD** rule: it solve the case $1//L_{max}$ too minimizes T_{max} for the $1//T_{max}$ case, this order of jobs in increasing di for all job i [22].

3. Moore's Algorithm(**MA**): that solve the case $1//\sum U_i$ an Algorithm (MA) following [1]:

   Step 0: arrangement job EDD rule, Let $F = H = \{\phi\}, g = z = 0$.

   Step 1: $g = g + 1$, if $g > n$ going to Step 3.

   Step 2: $z = z + p_k, F = FU\{g\},\ if\ \ z \leq d_k$ going to Step 1, except that (if$z > d_k$) now catch a job $j \in F$ has p_j as big as likely and consider $z = z - p_j, F = F - \{j\}, H = HU_{\{j\}}$ and going to step 1.

   Step 3: F collection of early jobs, and H collection of late jobs.

4. Lawler algorithm (**LA**) which solves the $1//f_{max}$ problem where $f_{max} \in \{C_{max}, L_{max}, T_{max}, V_{max}\}$. The algorithm of (LA) is presented in following [1]:

   Step 0: consider $M = \{1, 2, \ldots, m\}, \pi = (\phi)$ and G the collection of $\forall j$ with no successors.

   Step 1: consider $(j^\star)$ s.t $f_{j^\star}(\sum_{i\in M} P_i) = Min_{j\in G}\{f_j(\sum_{i\in M} P_i)\}$
   Collection $M = M - \{j^*\}$ and arrangement job j^* in latest location of π.
   Change G to act the new collection of scheduling.

   Step 2: if $M = \phi$ stop , if not going to Step 1.

3. Minimizing $(\sum U_i + \sum T_i + T_{max})$

Now keep in mind that $M = \{1, 2, \ldots, m\}$ are the jobs that each $j \in M$ needs a processing time Pj and a due date dj, which are done on single machine, so that give the sequences $\sigma = (\sigma(1), \ldots, \sigma(m))$ of jobs and our goal is to find the optimal solution for $(\sum U_i + \sum T_i + T_{max})$ on single machine.

- Some problem find minimum of the total number of late job $(\sum U_i)$.
- Some problem find The minimization of the total of tardiness $(\sum T_i)$.
- Some problem find The minimization of the maximum tardiness (T_{max}).

denoted by this problem is $1//F(\sum U_i, \sum Ti, T_{max})$ problem (P)
Try to find efficient (Pareto optimal) solutions for (P) which can be written as:

$$\left.\begin{array}{ll} \multicolumn{2}{c}{\text{Min}\begin{Bmatrix}\sum U_i\\ \sum T_i\\ T_{\max}\end{Bmatrix}} \\ \multicolumn{2}{c}{\text{s.t.}} \\ U_{\sigma(i)}=1 & \text{i=1},\ldots,n \\ U_{\sigma(i)}=0 & \text{i=1},\ldots,\text{n} \\ T_{\sigma(i)} \geq C_{\sigma(j)}-d_{\sigma(j)} & \text{i=1},\ldots,n \\ T_{\sigma(i)} \geq 0 & \text{i=1},\ldots,\text{n} \end{array}\right\} \ldots\ldots\ldots\ldots\ldots (P)$$

Problem (P) is one of the problems that is difficult to solve, so we find a set of acceptable solutions(pareto optimal solve), and we also suggest an algorithm that works to find these acceptable solutions to our problem.

particular problems for (P)

Now will show a set of partial problems, which are particular problems for (P), the objective is to finding order that minimize the multicriterian for the this cases:

1. $1//lex(\sum U_i, \sum T_i, T_{max})$ case (P1)
2. $1//lex(\sum U_i, T_{max}, \sum T_i)$ case (P2)
3. $1//lex(T_{max}, \sum U_i, \sum T_i)$ case (P3)
4. $1//lex(T_{max}, \sum T_i, \sum U_i)$ case (P4)
5. $1//lex(\sum T_i, \sum U_i, T_{max})$ case (P5)
6. $1//lex(\sum T_i, T_{max}, \sum U_i)$ case (P6)
7. $1//(\sum U_i + \sum Ti + T_{max})$ case (P7)

1 . $(1//Lex(\sum U_i, \sum Ti, T_{max}))$ case

The case can be written as

$$\left.\begin{array}{c} \text{Min } \text{T}_{\text{Max}} \\ \text{s.t.} \\ \sum_{i=1}^{n} U_i = \text{U}^\star \ \ , \text{U}^\star = \sum_{i=1}^{n} U_i \text{ (MA)} \\ \sum_{i=1}^{n} T_i \leq \text{T} \ \ , \text{T} \in [\sum_{i=1}^{n} T_i \text{(EDD)}, \ \sum_{i=1}^{n} T_i \text{ (MA)}] \end{array}\right\} \qquad \text{(P1)}$$

The problem (P1) with $\sum U_i$ is the most importance functional and must be optimum, , therefore the easy algorithm (AP1) Which give us the best result for (P1).

Algorithm (AP1) for case (P1)

Step 0: Job order by using Moore algorithm, get find late jobs and early jobs.

Step 1: Early job comes first and then late job comes after and we arrange all the jobs by using EDD rule get the schedule π

Step 2: Calculate $(\sum U_i, \sum Ti, T_{max})$ for the schedule π .

Example 3.1. *Consider the problem (P1) with the following data:* $p_i = (10, 5, 12, 2), d_i = (12, 16, 13, 9)$ *The Moore algorithm gives the late jobs (1,3) arrange them using EDD rule and early jobs (2,4) arrange them using EDD rule then get the sequence* $\pi = (4, 2, 1, 3)$, *then get* $(\sum U_i, \sum Ti, T_{max}) = (2, 21, 16)$ *by using the (AP1) for* $1//lex(\sum U_i, \sum T_i, T_{max})$ *case.*

2 . $(1//Lex(\sum U_i, T_{max}, \sum T_i))$ case

The case can be written as

$$\left.\begin{array}{c} \text{Min}\sum \text{ Ti} \\ \text{s.t.} \\ \sum_{i=1}^{n} U_i = \text{U}^\star \ \ , \text{U}^\star = \sum_{i=1}^{n} U_i \ (\text{MA}) \\ \text{T}_{\text{max}} \leq \ \text{T} \ \ , \ \text{T} \in [\ \text{T}_{\text{max}}\,(\text{LA})\,, \ \ \text{T}_{\text{max}} \ (\text{MA})] \end{array}\right\} \quad (\text{P2})$$

Since The problem (P2) with $\sum U_i$ is the most importance functional and must be optimum, , therefore (AP1) Which give us the best result for (P2).

3. $(1//Lex(T_{max}, \sum U_i, \sum Ti))$ case.

The case can be written as

$$\left.\begin{array}{c} \text{Min}\sum \text{ Ti} \\ \text{s.t.} \\ \text{T}_{\text{max}} = \text{T}^\star \qquad \qquad \text{T}^\star = \ \text{T}_{\text{max}}(LA) \\ \sum_{i=1}^{n} U_i \ \leq \ \text{U} \ \ , \ \text{U} \in [\sum_{i=1}^{n} U_i \ \ (\text{MA})\,, \ \sum_{i=1}^{n} U_i \ \ (\text{LA})] \end{array}\right\} \quad (\text{P3})$$

Since the problem (P3) with T_{max} is the most importance functional and must be optimum, therefore Lawler Algorithm (LA) Which give us the best result for (P3).

Example 3.2. *Consideration the case (P3) with the information:*$Pi = (1, 4, 3, 2), di = (6, 7, 4, 5)$ *The Lawler algorithm gives The sequence* $(3, 4, 1, 2)$ *with* $(T_{max}, \sum U_i, \sum Ti) = (3, 1, 3)$ *for the* $1//Lex(T_{max}, \sum U_i, \sum Ti)$ *problem.*

4. $(1//Lex(T_{max}, \sum T_i, \sum U_i))$ case.

The case can be written as

$$\left.\begin{array}{c} \text{Min}\sum U_i \\ \text{s.t.} \\ \text{T}_{\max} = \text{T}^{\star} \qquad \text{T}^{\star} = \text{T}_{\max}(LA) \\ \sum_{i=1}^{n} T_i \le \text{T} \quad , \text{T} \in [\sum_{i=1}^{n} T_i \ \ (\text{MA}) , \ \sum_{i=1}^{n} T_i \ \ (\text{LA})] \end{array}\right\} \qquad (\text{P4})$$

Since the problem (P4) with T_{max} is the most importance functional and must be optimum, therefore the Lawler Algorithm (LA) Which give us the best result for (P4).

5. $(1//Lex(\sum T_i, \sum U_i, T_{max}))$ case (P5) and 6.($1//Lex(\sum T_i, T_{max}, \sum U_i))$ case (P6):

The two cases (P5) and (P6) are (NP-hard cases), because $1//(\sum T_i)$ problem is (NP-hard) [3].

7. $(1//(\sigma U_i + \sigma T_i + T_{max}))$ case

The $1//\sigma U_i + \sigma T_i + T_{max}$ problem, which is special case of the problem $1//F(\sum U_i, \sigma T_i, T_{max})$. We can write it like this:

$$\left.\begin{array}{ll} \multicolumn{2}{c}{\min\{\Sigma U_{\delta(j)} + \Sigma T_{\delta(j)} + \text{T}_{\max}\delta(j)\ \}} \\ \multicolumn{2}{c}{\text{s.t.}} \\ U_{\delta(j)} = 1 & \text{j=1}, \ldots, m \\ U_{\delta(j)} = 0 & \text{j=1}, \ldots,\text{m} \\ T_{\delta(j)} \ge C_{\sigma(j)} - d_{\sigma(j)} & \text{j=1}, \ldots, m \\ T_{\delta(j)} \ge 0 & \text{j=1}, \ldots,\text{m} \end{array}\right\} \qquad (\text{P7})$$

The objective of this problem is to find the minimize value of the sum $(\Sigma U_{\delta(j)} + \Sigma T_{\delta(j)} + \text{T}_{\max}(\delta))$ which is getting through the best arrangement δ=(δ(1),. . . ,(δ(m)) for m jobs on a single machine where $\delta \in B$ ((where B set all possible solve) and can be minimize by using branch and bound method.

The special cases for (P7)

Case 1. If EDD rule give $C_i \le d_i \ \ \forall i \in N = \{1, \ldots, m\}$ so EDD scheduling given an optimum solution for the case (P7) equal the zero value.

Proof . as shown the EDD rule give $C_i \le d_i \ \ \forall i \in N \ so \ T_i = 0 \ \ \forall i \in N \ \ also \ \ \sum U_i = 0 \ \ \forall i \in N$ hence EDD order give optimum solution for (P7) equal zero value. □

Case 2. The EDD order is optimum solution for (P7) If$p_i = p \forall i \in N$ and $\sum U_i(EDD) = \sum U_i(MA)$.

Proof . as shown EDD order give least for T_{max} and if $Pi = P \forall i \in N$ so ((SPT)) and ((EDD)) are equal, then $\sum T_i$ is minimum and $\sum U_i EDD = \sum U_i(MA)$, therefore EDD order give an optimum solution for (P7). □

Decomposition of problem (P7)

Let $M = Min\{\sum U_i + \sum T_i + T_{max}\}$
The case (P7) can be divided to three sub-problems (S1), (S2) , (S3)

The schedule $\sigma \in S$(the set of allschedules)

$$\left.\begin{array}{ll} M_1 = \min \sum_{j=1}^{m} U_{\sigma(j)} & \\ U_{\sigma(j)} = \ 1 & j = 1, \dots, m \\ U_{\sigma(j)} = \ 0 & j = 1, \dots, m \end{array}\right\} \tag{S1}$$

$$\left.\begin{array}{ll} M_2 = \min \sum_{j=1}^{m} T_{\sigma(j)} & \\ T_{\sigma(j)} \geq \ \mathrm{C}_{\sigma(j)} \text{ - } \mathrm{d}_{\sigma(j)} & j = 1, \dots, m \\ T_{\sigma(j)} \geq 0 & j = 1, \dots, m \end{array}\right\} \tag{S2}$$

$$\left.\begin{array}{ll} M_3 = min\{T_{\max(\sigma(j))}\} & \\ T_{\sigma(j)} \geq \ \mathrm{C}_{\sigma(j)} \text{ - } \mathrm{d}_{\sigma(j)} & j = 1, \dots, m \\ T_{\sigma(j)} \geq & j = 1, \dots, m \end{array}\right\} \tag{S3}$$

This divided has the characteristics:
It is clear that (S1),(S2) and (S3) is simplest construction from the multicriterian case (P7), And it is simple to find optimality for S1and S3 by applying Moore algorithm and EDD order in that order and moreover to find a (LB) For (S2).

derivation (LB) and (UB) for (P7)

Now we start calculating (LB) and (UB) to solve the problem (P7) so that we can use the method branch and bound (BAB) to find the complete solution to this problem. The (LB) depends on solving three sub-problems(S1) ,(S2) and (S3) where we find (M1) which is the lower limit of problem (S1), and find (M2) which is the lower limit of problem (S2), and (M3) is the lower limit of (S3), and then we apply the following theorem:-

Theorem 3.3 ([1]). *$M_1 + M_2 + M_3 \leq M$ where M_1, M_2, M_3 and M are the smallest value of (S1),(S2),(S3) and (P7) in that order.*

To obtain (LB) for (P7). For (S1) calculation M1 by order the jobs in Moore algorithm through which we find the smallest total number of late job σU_i. For (S2) calculation (LB) for M2 by order the jobs in EDD) rule to find the smallest maximum tardiness.
Because $T_{max}(EDD) \leq \sum T_i(OPT)$. Hence $T_{max}(EDD)$ is a (LB) for M_2((i.e $T_{max}((EDD)) \leq S2$)) for (S3) calculation M3 by order the jobs in EDD to find smallest T_{max}. Now applying Theorem (1) obtain $(LB) = M_1 + T_{max}(EDD) + M_3$.
To compute (UB) sequence the jobs by EDD order obtain $UB = \sigma U_i + \sigma T_i + T_{max}$ by EDD rule.

Algorithm ((A)) for (P7)

Step 0 arrange job $j \in B = \{1, \dots, b\}$ using EDD rule and get the present sequence σ

Step 1 calculate $E = \sum T_j + \sum U_j + T_{max}$ for the σ

Step 2 calculate $\sum P_j = Q_j$ and $X_j = P_j + d_j \forall j \in B$.

Step 3 Calculation $m_j \forall j \in B$ where $m_j = Q_j - X_j \forall j \in B$

Step 4 take i^* so that $i^* = Min\{mj\} \forall j \in B$
Set $B = B - \{i^*\}$, order the job i^* in latest place in σ_1.

Step 5 If $B = \{\phi\}$ progress to Step 6 , else return Step 2.

Step 6 Calculation $E_1 = \sum T_j + \sum U_j + T_{max}$ for σ_1.

Step 7 If $E_1 \leq E$ get $\sigma = \sigma_1$ and $E_1 = E$ is the Best solve for (P7).

Can be us the algorithm ((A)) to solve ((P)) to find near optimal solve for ((P)).

Local Search and Genetic Algorithm

Define local search methods in simple words, which is that it is a set of procedures that searches for the best solution among a set of possible solutions to the problem to be solved, where we determine for each problem Stop criteria to ensure we get the best solution**[11]**

1. descent method (DM)

This simple algorithm used to find an approximate solution for many scheduling problems. The first solution is random, then try to improve that solution as an adjacent solve the problem is found depend several techniques. If newly solve is best than present solve, then changes it, otherwise our solve remains same and repeat the process until get the best solution. can show this with a simple algorithm [17].

Algorithm (DM)

Step 0: select first random solve the problem $m \in S$ where S the set of all solution for this problem

Step 1: generate $(\acute{m})$ neighborhood (m) (by swap or insert) and set $\alpha = f(\acute{m}) - f(m), if \quad \alpha < 0$, then get the set $m = \acute{m}$.

Step 2: if $f(\acute{m}) \geq f(m),\ \forall \acute{m} \in N(S)$, thereafter end; go to step 1

2. Simulated annealing(SA)

This method is one of the arithmetic methods capable of escaping from local minimum limits. That a random, local methods for source: Initially, in vicinity the current solve, the neighborhood is random selection. After that, the best-cost neighbors are chosen, and mean the best, meaning the least expensive, although the probability gradually decreases in the path implementing algorithms. However, a natures of the algorithm approaches optimal solutions under temperate conditions. can show how this algorithm works through the following steps[10, 21].

Algorithm (SA)

Step 0: Choice a first solution $m \in S, m^* = m$; choice a first Temperature $t_o > 0; K = 0, g = 1$

Step 1: Defined b; choose $\acute{m} \in N^*(m); \alpha = f(\acute{m}) - f(m);\ \ P(\alpha, tK) = Exp(-\alpha/tK);$
If $\alpha \leq 0$, $then\ m = \acute{m},\ and\ if\ f(m) < f(m^*),\ then\ m^* = m;\ else(\alpha > 0)$; If a random number of $[0, 1] \leq P(\alpha, tK)$, then $m = \acute{m}; g = g + 1,$

Step 2: If$g \leq b$ go to step 1 .

Step 3: update Temperature; $K = K + 1$; go to step 1 Pending suspension criteria have been met.

3. genetic algorithm (GA)

This algorithm mostly exploration and optimization method that action on a inhabitants of possible solution (initial solution) for a caseThe next step defined the construction of (GA) [20, 18]

Step 0: initialize The first inhabitants from which to start the solution can be randomly generated or can build it using specific methods to reveal the heuristic and obtain a suitable initial population.

Step 1: newly Inhabitants The present Inhabitants, create a newly Inhabitants group that differs from the first group using several genetic factors: transformation, choice, and intersection.

Step 2: finish Finally, the step end upon execution while the result is stable for a number of times.

Comparison Computational Results for (P7)

Now solving problem(P7)by using exact methods(complete enumeration method (CEM), branch and bounded (BAB)) and solve this problem by using local search methods(descent method DM, simulated annealing SA, genetic algorithm GA and algorithm (A)) The output of all methods programmed in Matlab [9] and Compare with other methods of solving for all N five examples are generated when (1 ≤ Pj≤ 10) , (1 ≤dj≤20) and $d_j \geq Max(p_j)$. These algorithms were tested on problem (P7) with (3,. . . ,5000) jobs.
For each given value of N, five parameter values are generated that produce five problems ∀N. The performance of the CEM for problem (P7) They are compared in 5 cases for each N with the local search algorithm (DM), (SA), (GA) and algorithm (A). where N=3,5,7,9 The problems were generated randomly.

The performance the BAB algorithm for the problem (P7) They are compared in 5 cases for each N with the local search algorithm (DM), (SA), (GA) and algorithm (A). when N=3,5,7,9 ,15 The problems were generated randomly.

The solutions we get by applying local search methods (DM), (SA), (GA) and algorithm (A) for (P7) When N=20,25,50,100,1000,. . . ,5000 Cases are randomly generated. The Table (1) which show the values of the algorithms after applying them to a number of examples and how often they give us the best value for each value n. where

Table 1: The performance of local search methods for the $(1//F(\sum U_i + \sum T_i + T_{max})$ problem

		value						time					
N	ex	CEM	BAB	DM	SA	GA	A	CEM	BAB	DS	SA	GA	A
	1	0	0	0	0	0	0	g	g	g	g	g	g
	2	10	10	10	10	10	10	g	g	g	g	g	g
3	3	26	26	26	26	26	26	g	g	g	g	g	g
	4	7	7	7	7	7	7	g	g	g	g	g	g
	5	5	5	5	5	5	5	g	g	g	g	g	g
no. the Best		5	5	5	5	5	5	5	5	5	5	5	5
	1	25	25	25	25	25	25	g	g	g	g	g	g
	2	31	31	31	31	31	31	g	g	g	g	g	g
5	3	66	66	66	66	66	66	g	g	g	g	g	g
	4	24	24	24	24	24	24	g	g	g	g	g	g
	5	66	66	66	66	66	66	g	g	g	g	g	g
no. the Best		5	5	5	5	5	5	5	5	5	5	5	5
	1	103	103	103	103	103	103	g	g	g	g	g	g
	2	89	89	89	89	89	89	g	g	g	g	g	g
7	3	96	96	96	96	96	96	g	g	g	g	g	g
	4	64	64	64	64	64	64	g	g	g	g	g	g
	5	91	91	91	91	91	91	g	g	g	g	g	g
no. the Best		5	5	5	5	5	5	5	5	5	5	5	5
	1	130	130	130	130	130	130	13.02	g	g	g	g	g
	2	173	173	173	173	173	173	13.01	g	g	g	g	g
9	3	125	125	125	125	125	125	12.91	g	g	g	g	g
	4	166	166	166	166	166	166	12.82	g	g	g	g	g
	5	143	143	143	143	143	143	12.86	g	g	g	g	g
no. the Best		5	5	5	5	5	5	0	5	5	5	5	5
	1	—	390	390	390	390	390	—	113.26	g	g	g	g
	2	—	326	326	326	351	326	—	112.9	g	g	g	g
15	3	—	364	364	364	364	364	—	113.21	g	g	g	g
	4	—	472	472	472	472	472	—	112.87	g	g	g	g
	5	—	393	393	393	433	393	—	112.34	g	g	g	g
no. the Best		—	5	5	5	3	5	—	0	5	5	5	5
	1	—	—	790	790	790	790	—	—	g	g	g	g
	2	—	—	631	630	631	630	—	—	g	g	g	g
20	3	—	—	762	762	831	762	—	—	g	g	g	g
	4	—	—	798	798	820	798	—	—	g	g	g	g
	5	—	—	798	798	798	798	—	—	g	g	g	g
no. the Best		—	—	4	5	2	5	—	—	5	5	5	5
	1	—	—	1394	1387	1507	1387	—	—	g	g	g	g
	2	—	—	1152	1152	1333	1152	—	—	g	g	g	g
25	3	—	—	916	914	914	914	—	—	g	g	g	g
	4	—	—	915	914	1055	914	—	—	g	g	g	g
	5	—	—	1229	1229	1354	1229	—	—	g	g	g	g
no. the Best		—	—	2	5	1	5	—	—	5	5	5	5
	1	—	—	5361	5356	5356	5356	—	—	g	g	g	g
	2	—	—	4002	4002	5018	4002	—	—	g	g	g	g
50	3	—	—	4516	4509	4509	4509	—	—	g	g	g	g
	4	—	—	4326	4312	5255	4312	—	—	g	g	g	g
	5	—	—	4774	4774	5823	4774	—	—	g	g	g	g
no. the Best		—	—	2	5	2	5	—		5	5	5	5
	1	—	—	19068	19120	24088	19068	—	—	g	g	g	g
	2	—	—	21013	21076	26835	21013	—	—	g	g	g	g
100	3	—	—	19567	19570	19567	19567	—	—	g	g	g	g
	4	—	—	21526	21520	26374	21520	—	—	g	g	g	g
	5	—	—	17831	18107	23266	17830	—	—	g	g	g	g
no. the Best		—	—	4	1	1	5	—	—	5	5	5	5
	1	—	—	2284761	2269827	2686875	2269827	—	—	1.9	1.9	g	g
	2	—	—	2367109	2350523	2755129	2350523	—	—	2	1.9	g	g
1000	3	—	—	2316250	2302234	2684700	2302234	—	—	1.9	1.9	4.7	g
	4	—	—	2268245	2260874	2658480	2260874	—	—	1.8	2	g	g
	5	—	—	2323225	2324233	2708127	2323233	—	—	1.9	2	4	g
no. the Best		—		1	4	0	5	—	—	0	0	3	5
	1	—	—	9480968	9399914	10863554	9399914	—	—	3.8	3.8	1	g
	2	—	—	9653610	9629966	9734234	9629966	—	—	3.9	3.8	2.3	g
2000	3	—	—	9539264	9521345	10755456	9521345	—	—	3.8	3.8	7.6	g
	4	—	—	9608018	9568179	9874269	9568179	—	—	3.9	3.9	10.3	g
	5	—	—	9568587	9575619	10701796	9565619	—	—	3.8	3.8	3.2	g
no. the Best		—		1	4	0	5	—	—	0	0	0	5
	1	—	—	61601671	61579007	67976291	61579007	—	—	9.3	9.4	5.9	g
	2	—	—	61037958	60945229	67502106	60945229	—	—	9.4	9.4	g	g
5000	3	—	—	62062746	61964617	68570540	61964617	—	—	9.4	9.4	3.7	g
	4	—	—	62300415	62452345	69180855	62300415	—	—	9.4	9.4	16.4	g
	5	—	—	60730544	60932613	62362495	60730544	—	—	9.4	9.3	25.9	g
no. the Best		—	—	2	3	0	5	—	—	0	0	1	5
Total no. best		20	25	41	52	29	60	15	20	45	45	49	60

N = numbers of job.

ex = Number of example.

DM = values found descent algorithm.

SA = the value found by simulation annealing.

GA = the value found by Genetic Algorithm (GA).

A = the value found by Algorithm (A).

no. the Best = Number of examples that give us the best value.

Time = in second and $0 \leq g \leq 1$ second.

4. Conclusion and future work

In this research, with total late number job, total tardiness and maximum tardiness has been considered. The result of table (1) shows that the algorithm(A) performs very well and then come (SA)and then come (DA) which gives reasonable results . It is clear that the time values for both DM and SA algorithm are quite similar But the algorithm(A) is better than them .An interesting future research topic would involve experimentation with exact and local algorithm for the following problems

1. $1/\text{ri}/1/(\sum \text{Ci}\ , \sum \text{Ti}, T_{max})$
2. $1/\text{ri}/1/(\sum \text{Ci}\ , \sum Ei, Emax)$

References

[1] D. A. Abbass, *Using Branch and Bound and Local Search Methods to Solve Multi-objective Machine Scheduling Problem*, IEEE SmartWorld Ubiquitous Intell Comput. (2019) 63–66.

[2] I. T Abbas, *The Performance of Multicriteria Scheduling in one Machine*, M.Sc tThesis, Univ. of Al-Mustansiriyah, College of Science, Dept. of Mathematics, 2009.

[3] T. S. Abdul-Razaq, Z. M. Ali, *Minimizing the Total Completion Time, the Total Tardiness and the Maximum Tardiness*, Ibn Al-Haitham J. Pure Appl. Sci. 28(2) (2015) 155–170.

[4] T.S. Abdul-Razaq and A.O. Akram, *Local search algorithms for multi-criteria single machine scheduling problem*, Ibn Al-Haitham J. Pure Appl. Sci. (2017) 436-451.

[5] M.G. Ahmed, *A single machine scheduling problem to minimize the sum of total Completion times and total Late works*, Res. Al-Mustansiriyah J. Sci. 23(7) (2012) 117–130.

[6] H. Ali, *Exact and Heuristic Algorithms for Solving Combinatorial Optimization Problems*, PH.D. Thesis, Department of Mathematics, College of Science, University of Al-Mustansiriyah, 2017.

[7] S.S. Al-Assaf, *Solving Multiple Objectives Scheduling Problems*, M.Sc. Thesis, Univ. of Al-Mustansiriyah, College of Science, Dept. of Mathematics, 2007.

[8] M.K. Al Zuwaini, S.K. Al Saidy and T. S. Abdul-Razaq, *Comparison study for some local search methods for multiple objective function in a single machine scheduling problem*, J. Basrah Res. (Science) 37(4) (2011) 103–113.

[9] N. Anang, M.S. Hamid and W.M.W. Muda, *Simulation and modelling of electricity usage control and monitoring system using rhing speak*, Baghdad Sci. J. 18 (2021) 907–927.

[10] H. Chachan and A. Hameed, *Exact methods for solving multi objective problem on single machine scheduling*, Iraqi J. Sci. 60 (2019) 1802–1813.

[11] A. Gupta and R.P. Mahapatra, *Multifactor algorithm for test case selection and ordering*, Baghdad Sci. J. 18 (2021) 1056–1075.

[12] J.A. Hoogeveen, *Multicriteria Scheduling*, Eur. J. Oper. Res. 167 (2005) 592–623.

[13] C. Junheng, C. Feng, L. Ming, W. Peng and X. Weili, *Bi-criteria single machine batch scheduling with machine on/off switching under time-ofuse tariffs*, Comput. Ind. Eng. 112 (2017) 721–734.

[14] M.E. Kurz and S. Canterbury *Minimizing total flow time and maximum earliness on a single machine using multiple measures of fitness*, Genetic and Evolutionary Computation Conf. (2005) pp. 803-809.

[15] A.A. Mahmood, *Approximation solution for multicriteria scheduling*, Prob. Res. Al-Rafidain University College Sci. 1(34) (2014) 161–179.

[16] A.A. Mahmood and T.S. Abdul-Razaq, *Exact algorithm for minimizing the sum of total late work and maximum late work problem*, Res. Diyala J. Pure Sci. 10 (2014) 39–50.

[17] A.A. Mahmood Al-Nuaimi, *Local search algorithms for multiobjective scheduling problem*, Al Rafidain J. University College Sci. (1681-6870) 2015201-217.

[18] A. Muthiah, R. Rajkumar and B. Muthukumar, *Minimizing makespan in job shop scheduling problem using genetic algorithm*, Appl. Mech. Mater. 813-814 (2015) 1183–1187.

[19] E.O. Oyetunji and A.E. Oluleye, *Heuristics for minimizing total completion time and number of tardy jobs simultaneously on single machine with release time*, Res. J. Appl. Sci.3 (2008) 147–152.

[20] M. Reisi-Nafchi and G.A. Moslehi, *Hybrid genetic and linear programming algorithm for two-agent order acceptance and scheduling problem*, Appl. Soft. Comput. 33 (2015) 37–47.

[21] W. Shao and Z. Shao, *A Pareto-based estimation of distribution algorithm for solving multiobjective distributed no-wait flow-shop scheduling problem with sequence dependent setup time*, IEEE. Trans. Autom. Sci. Eng. 16 (2019) 1344–1360.

[22] N. Tyagi, R.P. Tripathi and A.B. Chandramouli, *Single machine scheduling model with total tardiness problem*, Indian J. Sci. Technol. 9(37) (2016) 1–14.