 https://doi.org/ 10.30526/2017.IHSCICONF.1817

Local Search Algorithms for Multi-Criteria Single Machine Scheduling Problem

Tariq S. Abdul-Razaq
Abeer O. Akram
abeeromar1985@gmail.com
Dept. of Mathematics/ College of Science / University of Al-Mustansiriyah

Abstract

Real life scheduling problems require the decision maker to consider a number of criteria before arriving at any decision. In this paper, we consider the multi-criteria scheduling problem of n jobs on single machine to minimize a function of five criteria denoted by total completion times ($\sum C_i$), total tardiness ($\sum T_i$), total earliness ($\sum E_i$), maximum tardiness (T_{max}) and maximum earliness (E_{max}). The single machine total tardiness problem and total earliness problem are already NP-hard, so the considered problem is strongly NP-hard. We apply two local search algorithms (LSAs) descent method (DM) and simulated annealing method (SM) for the 1// ($\sum C_i + \sum T_i + \sum E_i + T_{max} + E_{max}$) problem (SP) to find near optimal solutions. The local search methods are used to speed up the process of finding a good enough solution, where an exhaustive search is impractical for the exact solution. The two heuristic (DM and SM) were compared with the branch and bound (BAB) algorithm in order to evaluate effectiveness of the solution methods.

Some experimental results are presented to show the applicability of the (BAB) algorithm and (LSAs). With a reasonable time, (LSAs) may solve the problem (SP) up to 5000 jobs.

Keywords: Multicriteria; Scheduling; Single machine; Earliness-tardiness; local search methods.

 https://doi.org/ 10.30526/2017.IHSCICONF.1817

1. Introduction

Scheduling is allocation of resources (machines) over time to perform a collection of tasks (jobs).

Generally speaking, Scheduling means to assign machines to jobs in order to complete all jobs under the imposed constraints. The problem of scheduling a set N ={1,…,n} of n jobs on a single machine. Each job i∈N has processing time p_i and a due date d_i. If a given schedule σ= (1,…,n), then the completion time $C_i=\sum_{j=1}^{i} p_j$, the tardiness of job i T_i=*max*{c_i-d_i,0} and earliness of job i E_i=*max*{d_i-c_i,0}, consequently we have total completion time $\sum_{i\in N} C_i$, total tardiness $\sum_{i\in N} T_i$, maximum tardiness $T_{max}=\max_{i\in N}\{T_i\}$, total earliness $\sum_{i\in N} E_i$ and maximum earliness $E_{max}=\max_{i\in N}\{E_i\}$.

For the maximum tardiness for *1//*T_{max} problem is minimized by EDD (earliest due date) rule to Jackson 1955[9]. The *1//*$\sum C_i$ problem, the (SPT) (shortest processing time) rule is optimal to Smith 1956[13]. The maximum earliness for *1//*E_{max} problem is minimized by MST (minimum Slack time) rule [3], where the two problems *1//*$\sum T_i$ and *1//*$\sum E_i$ are NP-hard ([6],[11]) and [3] respectively. Any problem including such cost functions as subproblem is NP-hard.

The first bi-criteria scheduling problem was already solved by Smith (1956) [13] the *1//*($\sum C_i, T_{max}$) problem subject to T_{max} =0 is imposed by using back ward algorithm, only a few bi-criteria scheduling problem have been investigated since then. Van Wassenhove & Gelder (1980) [16] studied the *1//*($\sum C_i, T_{max}$) problem. The set of efficient points is characterized and a pseudo-polynomial algorithm to enumerate all these points is given.

Hoogveen and Van de velde (1995)[8] provided an algorithm for finding all efficient schedules for the problem *1//*($\sum C_i, f_{max}$). Tadie et al. (2002) [15] proposed a procedure that takes advantage of an algorithm for finding the Pareto optima set by applying specially developed constraints to a branch and bound (BAB) algorithm for the *1//*(ΣT_i, T_{max}) problem to find the set of efficient point. For the *1//*(ΣC_i, E_{max}) problem, Kurz and Canterbury (2005) [10] used genetic algorithm, Al-Assaf (2007)[5] proposed BAB algorithm to find the optimal solution for *1//*$\sum C_i + E_{max}$ problem and proposed an algorithm with a special range for the problem *1//*(ΣC_i, E_{max}) to find the set of efficient solutions.

The single machine *1//*$\sum C_i + \sum T_i + T_{max}$ problem is NP-hard, the (BAB) algorithm is used to find optimal solution (2015)[1]. For *1//*$\sum C_i + \sum E_i + E_{max}$ problem is NP-hard, local search algorithms are used to find near optimal solution and compared their results with CEM for small n (2016) [2]. There are mainly three classes of approaches that are applicable to multicriteria scheduling problem.

$\mathbf{C_1}$**: Hierarchical (lexographical)** optimization the hierarchical approach, one of the criteria (more important) regards as constraint (primary) criterion which must be satisfied, (see [7] and [14]).

 https://doi.org/ 10.30526/2017.IHSCICONF.1817

C_2: Priority optimization

In this approach minimizing a weighted sum of the multicriteria (objectives) and convert the multicriteria to a single criterion problem, several multicriteria scheduling problems studied in this class (see [8]and [12]).

C_3: Interactive optimization

In this approach one generates all efficient (Pareto optimal) schedules and select the one that yield the best composite objective function value of the multicriteria. Several multicriteria scheduling problems studied in this class (see [8] and [16]).

2. Problem Formulation and Analysis

We consider the following performance criteria: $\sum_{i\in N} C_i$, $\sum_{i\in N} T_i$, $\sum_{i\in N} E_i$, T_{max} and E_{max} hence the problem is denoted by *1//F* $(\sum C_i, \sum T_i, \sum E_i, T_{max}, E_{max})$ (P). We consider multicriteria problem of scheduling n jobs on a single machine. All jobs are available at time zero and characterized by their processing time p_i and due date d_i. In this problem, the total completion times (total flow times), the total tardiness, the total earliness, maximum tardiness and maximum earliness are used as multicriteria. The first object is to minimize flow time (a measure for average in processing inventory). The other objectives deal with service to customers. These objective functions force jobs not be early and/or tardy.

For this problem, we will try to find efficient solutions for the *1//F* $(\sum C_i, \sum T_i, \sum E_i, T_{max}, E_{max})$ problem (P), which can be written for a given schedule s= (1,…,n) as:

$$\left.\begin{array}{ll}
\underset{s\in S}{Min}\begin{cases}\sum C_i(s)\\ \sum T_i(s)\\ \sum E_i(s)\\ T_{max}(s)\\ E_{max}(s)\end{cases} & \\
s.t & \\
C_i \geq P_i & i=1,\dots,n\\
C_i = C_{(i-1)} + P_i & i=2,\dots,n\\
T_i \geq C_i - d_i & i=1,\dots,n\\
T_i \geq 0 & i=1,\dots,n\\
E_i \geq d_i - C_i & i=1,\dots,n\\
E_i \geq 0 & i=1,\dots,n
\end{array}\right\} \dots (P)$$

Where **S** is the set of all schedules.

This problem (P) is difficult to solve and find the set of all efficient solutions (SE). This problem of five objects has not been considered by any researcher yet. We propose efficient algorithm to find approximate set of efficient solutions (SA) for this problem.

1- Some results for the *1//F* $(\sum C_i, \sum T_i, \sum E_i, T_{max}, E_{max})$ **problem (P):**

Proposition (1): The SPT sequence is efficient for the problem (P).

Proof: First, suppose that all processing times are different the unique SPT sequence (SPT^*) gives the absolute minimum of $\sum C_i$. Hence there is no sequence $\sigma \neq SPT^*$ such that

 https://doi.org/ 10.30526/2017.IHSCICONF.1817

$\sum C_i(\sigma) \leq \sum C_i(SPT^*), \sum T_i(\sigma) \leq \sum T_i(SPT^*), \sum E_i(\sigma) \leq \sum E_i(SPT^*),$

$T_{max}(\sigma) \leq T_{max}(SPT^*)$ and $E_{max}(\sigma) \leq E_{max}(SPT^*)$ … (3.1)

With at least one strict inequality.

Second if more than one SPT sequence exists, jobs with equal processing times are order in EDD rule, if SPT and EDD are identical then order these jobs in MST and let the resulting SPT sequence (SPT^*). Note if σ is an SPT but not SPT^* sequence it can not dominate SPT^* sequence since:

$\sum C_i(\sigma) = \sum C_i(SPT^*), \sum T_i(SPT^*) \leq \sum T_i(\sigma), \sum E_i(SPT^*) \leq \sum E_i(\sigma),$

$T_{max}(SPT^*) \leq T_{max}(\sigma)$ and $E_{max}(SPT^*) \leq E_{max}(\sigma)$ … (3.2)

Hence SPT^* sequence is efficient.■

Proposition (2): If SPT rule, EDD rule and MST rule are identical, then there is one or more than one efficient solution for the problem (P).

Proof: It is clear that this identical sequence (s) is efficient by proposition (1). Now since $\sum E_i$ is non regular criteria, there may be another sequence (s') with value of $\sum E_i(s') \leq \sum E_i(s)$. Hence the sequence s' is also efficient solution■.

Example (1): consider the problem (P) with the following data: Pi=(2,3,3,5) , di=(3,6,8,10) and Si=(1,3,5,5). The sequence (1,2,3,4) is SPT, EDD and MST give the only one efficient. ($\sum C_i, \sum T_i, \sum E_i, T_{max}, E_{max}$)= (28,3,2,3,1), which is obtained by (CEM).

Proposition (3): If SPT rule and MST rule are identical then there is one or more than one efficient solution.

Proof: The sequence s=(1,…,n) obtained from the identical SPT rule and MST rule respectively.

Hence we have:

$P_1 \leq P_2 \leq \ldots \leq P_n$ …(3.3)

$d_1 - P_1 \leq d_2 - P_2 \leq \ldots \leq d_n - P_n$ …(3.4)

The EDD rule $d_1 \leq d_2 \leq \ldots \leq d_n$ is obtained by adding (3.3) & (3.4)

Hence the SPT, EDD and MST are identical, and we have one or more than one efficient solution by proposition (2) ■.

2- Algorithm (AP) for Determination of Approximate Set of Efficient Solutions for the Problem (P).

We propose algorithm (AP) to determine the set of approximate solutions (SA) for the problem (P).

This algorithm consists of two parts, the first part deals with calculation of tardiness and total completion times, the second part deals with calculation of earliness and total completion times.

 https://doi.org/ 10.30526/2017.IHSCICONF.1817

Algorithm (AP)for finding efficient solutions for the problem ***1//***$(\sum C_i, \sum T_i, \sum E_i, T_{max}, E_{max})$ **(P) :**

Step(0): Set $\Delta=\sum P_i$ and $\sigma = (\emptyset)$.

Step(1): Set $N=\{1,\ldots,n\}$,$K=n$,$t=\sum P_i$.

Step(2): Calculate T_i $\forall$ i $\in$N (by lawler algorithm).

Step(3): Find a job j$\in$N such that $T_j \leq \Delta$,$P_j \geq P_i$ $\forall j, i \in N$ and $T_i \leq \Delta$ assign job j in position K of σ if no job j with $T_j \leq \Delta$, set $E_{max}(\sigma) = E_{max}(spt)$ go to step(7).

Step(4): Set $t=t-P_j$,$N=N-\{j\}$,$K=K-1$,if K>1go to step (2).

Step(5): for the resulting sequence job $\sigma = (\sigma(1), \ldots \sigma(n))$ calculate $(\sum C_i(\sigma), \sum T_i(\sigma), \sum E_i(\sigma), T_{max}(\sigma), E_{max}(\sigma))$.

Step(6): Put $\Delta = T_{max}(\sigma)-1$,go to step(2).

Step(7): Put $\Delta = E_{max}(\sigma)-1$,$N=\{1,\ldots,n\}$,$K=1$,$t=\sum P_i$ and $\sigma = (\emptyset)$ if $\Delta < E_{max}$ (MST) go to step(11).

Step(8): Calculate $r_i = max\{s_i - \Delta, 0\}$ $\forall$ i $\in$N.

Step(9): Find a job j $\in$N with min r_j , $r_j \leq C_{K-1}$ and $P_j \leq P_i$ $\forall$j, i $\in$N, $C_0=0$ (break tie with small s_j) assign j in position K of σ.

Step(10): Set N=N-{j} ,K=K+1 ,if K $\leq$ n go to step(9) for the ruslting Sequence $\sigma = (\sigma(1), \ldots \sigma(n))$ calculate $(\sum C_i(\sigma), \sum T_i(\sigma), \sum E_i(\sigma), T_{max}(\sigma), E_{max}(\sigma))$ and go to step(7).

Step(11): Stop with a set of efficient solutions (SA).

Example (2): consider the problem (P) with the following data:
Pi=(3,4,8,7) , di=(12,4,10,7).
The result of efficient solutions for example (2) by CEM and algorithm AP.

Efficient solutions for problem (P)							
CEM	Alg.(AP)	$\sum C_i$	$\sum T_i$	$\sum E_i$	T_{max}	E_{max}	sum
(1,2,4,3)	(1,2,4,3)	46	22	9	12	9	98
(2,4,1,3)	(2,4,1,3)	51	18	0	12	0	81
(2,4,3,1)	(2,4,3,1)	56	23	0	10	0	89
(2,1,4,3)	(2,1,4,3)	47	19	5	13	5	89

In this example we find all efficient schedules, and sum is the optimal sum of $(\sum C_i(\sigma), \sum T_i(\sigma), \sum E_i(\sigma), T_{max}(\sigma), E_{max}(\sigma))$ =81

3- Sub-Problems of the Multicriteria Problem (P)

Decomposition of the problem (P) is a general approach for solving a problem by breaking it up into smaller ones. It is clear that this decomposition has the following properties:
First all the subproblems have simpler structure than the multicriteria problem (P). Second all the subproblems are NP-hard (except (P2) and (P3) are solved by pseudo algorithms) and some of them are studied by some researchers, such as (P4, p7, P8, P12, P13, P18, P19)

From the problem P we can get the following subproblems:

 https://doi.org/ 10.30526/2017.IHSCICONF.1817

1) $1//(\sum E_i, T_{max}, E_{max})\ldots P1$
2) $1//(\sum C_i, T_{max})\ldots P2$
3) $1//(\sum C_i, E_{max})\ldots P3$
4) $1//(\sum C_i, \sum T_i)\ldots P4$
5) $1//(\sum C_i, \sum E_i)\ldots P5$
6) $1//(\sum T_i, \sum E_i)\ldots P6$
7) $1//(T_{max}, E_{max})\ldots P7$
8) $1//(\sum T_i, T_{max})\ldots P8$
9) $1//(\sum E_i, E_{max})\ldots P9$
10) $1//(\sum E_i, T_{max})\ldots P10$
11) $1//(\sum T_i, E_{max})\ldots P11$
12) $1//(\sum T_i, \sum E_i, T_{max}, E_{max})\ldots P12$
13) $1//(\sum C_i, \sum T_i, T_{max}, E_{max})\ldots P13$
14) $1//(\sum C_i, \sum E_i, T_{max})\ldots P14$
15) $1//(\sum C_i, \sum T_i, E_{max})\ldots P15$
16) $1//(\sum C_i, \sum T_i, \sum E_i)\ldots P16$
17) $1//(\sum C_i, T_{max}, E_{max})\ldots P17$
18) $1//(\sum C_i, \sum T_i, T_{max})\ldots P18$
19) $1//(\sum C_i, \sum E_i, E_{max})\ldots P19$
20) $1//(\sum T_i, T_{max}, E_{max})\ldots P20$

For the sub-problems from (P13 to P17) we can use (AP) to find approximate set of efficient solutions.

4- The $1//(\sum C_i + \sum T_i + \sum E_i + T_{max} + E_{max})$ Problem (SP)

It is clear that the problem (SP) is a special case of the problem (P). The aim of this problem is to find the minimum value of the objective function $\sum C_i + \sum T_i + \sum E_i + T_{max} + E_{max}$. This problem is NP-hard and local search algorithms are used to find its optimal solution. This problem can formally be written for a given schedule s=(1,…,n) as:

$$\left.\begin{array}{ll} V = min\{\sum C_i + \sum T_i + \sum E_i + T_{max} + E_{max}\} & \\ s.t. & \\ C_i \geq p_i & i = 1, \ldots, n \\ C_i = C_{(i-1)} + p_i & i = 2, \ldots, n \\ T_i \geq C_i - d_i & i = 1, \ldots, n \\ T_i \geq 0 & i = 1, \ldots, n \\ E_i \geq d_i - C_i & i = 1, \ldots, n \\ E_i \geq 0 & i = 1, \ldots, n \end{array}\right\} \ldots \text{(SP)}$$

The aim for problem (SP) is to find a processing order $\sigma=(\sigma(1),\ldots,\sigma(2))$ of the jobs on a single machine to minimize the sum of the total completion time, total tardiness, total earliness, the maximum tardiness and the maximum earliness $(\sum C_{\sigma(i)} + \sum T_{\sigma(i)} + \sum E_{\sigma(i)} + T_{max}(\sigma) + E_{max}(\sigma))$, for a particular schedule $\sigma \in S$ where S is the set of all feasible solutions.

 https://doi.org/ 10.30526/2017.IHSCICONF.1817

3. Computational Experiments

3.1 Test problems

Performance of the algorithm (AP) for the problem (P) is compared on 5 problem instances for each n with the complete enumeration method (CEM). For each job j, the processing time p_j was uniformly generated from uniform distribution [1,10]. Also, for each job j, an integer due date d_j is generated from the uniform distribution [(1-TF-RDD/2)TP,(1-TF+RDD/2)TP], where TP is the total processing times of all the jobs, TF is the tardiness factor, and RDD is the relative range of the due dates. For the two parameters TF and RDD, the values 0.2,0.4,0.6,0.8,1.0 for TF and the values 0.9,1.0 for RDD are considered. For each selected value of n, one problem is generated for each of the five values of parameter producing 5 test problems.

3.2 Computational results for the problem (P)

In the Table (1) and Table (2) we have:
n: Number of jobs
EX: Example number
|CEM|: The cardinal number (exact number) of efficient solutions obtained by Complete Enumeration method (CEM).
|Alg AP|: The cardinal number (approximate number) of efficient solutions obtained by algorithm (AP).
Optimal: The optimal value of sum of $(\sum C_i, \sum T_i, \sum E_i, T_{max}, E_{max})$ obtained by (CEM).
Best: The optimal or near optimal of sum of $(\sum C_i, \sum T_i, \sum E_i, T_{max}, E_{max})$ obtained by algorithm (AP), for n≤10 and 11≤n≤100 respectively.

 https://doi.org/ 10.30526/2017.IHSCICONF.1817

Table (1): Comparison of number of efficient solutions of the algorithm (AP) with (CEM) for n≤10 for the problem (P).

Number of Efficient solutions for (P)							
n	EX	|CEM|	Optimal	Time	|Alg AP|	Best	Time
4	1	8	108	0.1197	5	108	0.2221
	2	4	138	0.0126	4	138	0.0393
	3	9	155	0.0127	4	155	0.0813
	4	3	83	0.0071	2	83	0.669
	5	10	59	0.0151	6	59	0.0107
			No. of opt.			5	
5	1	18	85	0.1270	3	85	0.1340
	2	13	140	0.0299	7	140	0.2027
	3	43	158	0.0432	8	158	0.0401
	4	49	107	0.0463	11	109	0.0542
	5	8	121	0.0227	6	121	0.1525
			No. of opt.			4	
6	1	41	156	0.2072	8	156	0.1808
	2	16	152	0.0953	3	152	0.0171
	3	52	197	0.1189	7	197	0.0322
	4	26	193	0.1147	8	193	0.138
	5	40	148	0.1119	4	148	0.0394
			No. of opt.			5	
7	1	64	232	9.4264	10	232	0.1539
	2	50	206	16.9584	3	224	0.1954
	3	58	246	10.7295	11	246	0.0499
	4	59	226	12.2096	9	226	0.0149
	5	55	189	88.6240	10	191	0.0605
			No. of opt.			3	
8	1	112	279	4.7087	1	299	0.1011
	2	132	265	4.4248	3	294	0.2308
	3	81	469	4.7034	1	518	0.0054
	4	73	343	4.7177	6	343	0.0495
	5	67	242	4.9829	8	242	0.0311
			No. of opt.			2	
9	1	74	295	39.5168	7	306	0.3677
	2	231	249	49.1618	10	264	0.1849
	3	139	384	44.8974	5	493	0.1562
	4	150	325	45.5338	2	375	0.0057
	5	205	344	43.6530	4	381	0.2455
			No. of opt.			0	
10	1	345	394	493.0906	19	394	0.3128
	2	173	292	469.6615	13	294	0.2171
	3	129	498	497.8566	7	498	0.0446
	4	398	414	461.0331	5	454	0.1882
	5	126	469	485.5750	17	469	0.2051
			No. of opt.			3	

Table (2): The results of approximate efficient solutions obtained by using algorithm (AP) when 11≤n≤100 for the problem (P).

Number of efficient solutions for (P)									
N	EX	Best	|Alg AP|	Time	n	EX	Best	|Alg AP|	Time
11	1	539	19	0.1168	12	1	627	24	0.1468
	2	676	14	0.0254		2	769	16	0.0989
	3	845	17	0.0302		3	769	3	0.0336
	4	429	8	0.0265		4	639	22	0.0400
	5	423	26	0.0457		5	740	2	0.0966
13	1	718	22	0.1542	14	1	1054	16	0.1131
	2	614	33	0.0626		2	944	25	0.0491
	3	761	35	0.0655		3	929	5	0.0277
	4	850	21	0.0392		4	652	24	0.0894
	5	776	9	0.0988		5	1002	10	0.0237
15	1	1163	31	0.1629	20	1	2243	12	0.4208
	2	1437	5	0.0126		2	1982	50	0.1459
	3	1203	2	0.1059		3	2307	41	0.1064
	4	1167	3	0.1121		4	1695	2	0.1548
	5	1221	34	0.0691		5	1583	47	0.1290
25	1	3405	2	0.2792	30	1	4733	7	0.1291
	2	3701	50	0.1946		2	4537	7	0.1904
	3	3782	2	0.1534		3	4517	50	0.1743
	4	3630	2	0.1608		4	4466	50	0.1791
	5	2937	50	0.1486		5	4477	2	0.2007
40	1	8553	50	0.3199	50	1	12541	5	0.3804
	2	8438	50	0.2256		2	12816	2	0.2687
	3	10368	50	0.2138		3	13280	7	0.2838
	4	7863	2	0.2351		4	9580	50	0.2662
	5	6729	50	0.2472		5	11708	45	0.2587
75	1	27124	14	0.5125	100	1	49021	2	0.6389
	2	25069	17	0.3946		2	47162	2	0.5283
	3	27450	2	0.4125		3	54565	2	0.5105
	4	25915	2	0.3821		4	47173	50	0.5185
	5	25514	16	0.4268		5	45498	2	0.5025

Note from the Table (1) the results show that:
For 4≤ n ≤10 the algorithm (AP) gives 22 exact optimal solutions for the problem (SP) from 35 test problems.

From the results of Tables (1) and (2) it is clear that the algorithm (AP) does not give good results for problems with large n. This is because the Multicriteria scheduling problems are generally affected by a number of costs functions, and in our problems (P) and (SP) the number of cost function is five.

Basic Structure of Local Search

For a Machine Scheduling problem

Given:

- Finite set S of feasible solutions
- Objective function f:S→R

The goal is to find a solution with a minimal objective value, i.e. a solution $s^* \in S$ with $f(s^*)=\min_{s\in S}\{f(s)\}$

Basic structure of Local Search Algorithm (LSA)

- Choose an initial solution;
- Repeat

Choose a solution from the neighborhood of the current solution and move to this solution

- Until stopping criteria

Variable Neighborhood Search (VNS) Algorithms

The (VNS) algorithms (DM and SM algorithms) depend on the selection of neighborhoods and the selection of the initial solution. In these(VNS) algorithms, we use three initial solutions s_1, s_2 and s_3 are obtained by solving the three single objective problems $1//\sum C_i$, $1//T_{max}$ and $1//E_{max}$ respectively.

The adjacent pair interchange (API) neighborhood (N) is used to generate new solutions. For the (VNS) algorithms, in each iteration initial solution s is selected, neighbor solutions are generated using N(s). The two algorithms (DM) and (SM) are run with stopping criterion at a known number of iterations depends on the number of jobs. Hence, we assign more iterations to large instances which are obviously more time consuming to solve.

Problem Instances

The performance of the DM and SM algorithms are compared on 5 problems instances. To compare the solutions that the sizes of these instances are:
for small size n=4,…,15
for middle size n=20,…, 150
for large size n=200,…5000

3.3 Computational Results for the Problem (SP)

Computational results of local search algorithms (LSAs) DM and SM is given in the following tables. We implement LSAs as follows: Since we know the optimal solutions for small size problems, which are obtained by BAB algorithm for n≤15[4], LSAs use large number of iterations, hence each algorithm stop when it catches the optimal solution (termination condition), but may be for large size problems we used 100000 iterations as termination condition. In these LSAs the neighborhoods generated using the API. The initial solution for the tested problems is generated using the minimum of (s_1,s_2,s_3).

The results obtained by LSAs is given in table (3). The results show which local search algorithm gives solution closed to optimal solution obtained by BAB and the corresponding time it needs to reach this solution for n≤15.

 https://doi.org/ 10.30526/2017.IHSCICONF.1817

In table (4) we give the results of comparison between LSAs themselves, for each algorithm, we find the best values and computation time. In Table (3), Table (4) and Table (5) we have:

n: Number of jobs
EX: Example number
Node: The number of nodes.
Optimal: The optimal value obtained by BAB algorithm [4].
No. of opt.:Number of examples that catch the optimal value.
No. of best: Number of examples that catch the best value.
SM:The value obtained by Simulation Annealing method.
DM:The value obtained by Decent Method.
Time: Time in seconds.

Table (3): The comparison between the optimal solutions obtained by BAB and the results of LSAs for small size problems

BAB					Local search			
N	EX	Optimal	Node	Time	DM	Time	SM	Time
4	1	108	11	0.0718	108	0.3366	108	0.3460
	2	138	7	0.0105	138	0.3310	138	0.3307
	3	155	7	0.0083	155	0.3216	155	0.3338
	4	83	7	0.0095	83	0.3241	83	0.3317
	5	59	14	0.0125	59	0.3289	59	0.3331
		No. of opt.			5		5	
5	1	85	23	0.0174	85	0.3352	85	0.3384
	2	140	19	0.0145	140	0.3257	140	0.3330
	3	158	49	0.0242	158	0.3239	158	0.3320
	4	107	94	0.0487	107	0.3234	107	0.3311
	5	121	14	0.0142	121	0.3239	121	0.3336
		No. of opt.			5		5	
6	1	156	32	0.0361	156	0.3297	156	0.3424
	2	152	30	0.0180	152	0.3248	152	0.3351
	3	197	63	0.0297	197	0.3234	197	0.3410
	4	193	67	0.0260	193	0.3252	193	0.3357
	5	148	28	0.0171	148	0.3277	148	0.3316
		No. of opt.			5		5	
7	1	232	266	0.0984	232	0.3355	232	0.3406
	2	206	310	0.0679	206	0.3544	206	0.3341
	3	246	105	0.0338	246	0.3375	246	0.3324
	4	226	173	0.0893	226	0.3291	226	0.3347
	5	189	155	0.0570	189	0.3376	189	0.3320
		No. of			5		5	

		opt.						
8	1	279	279	0.3884	279	0.3310	279	0.3445
	2	265	1966	0.9219	265	0.3473	265	0.3350
	3	469	206	0.0217	469	0.3320	469	0.3343
	4	343	90	0.0422	343	0.3301	343	0.3360
	5	242	181	0.0874	242	0.3285	242	0.3430
		No. of opt.			5		5	
9	1	295	878	0.3527	295	0.3330	295	0.3590
	2	249	1417	1.3732	249	0.3270	249	0.3387
	3	384	441	0.0681	384	0.3299	384	0.3470
	4	325	1571	1.3764	325	0.3271	325	0.3472
	5	344	3328	3.7414	344	0.3282	344	0.3414
		No. of opt.			5		5	
10	1	394	36196	41.3435	394	0.3318	394	0.3406
	2	292	1107	1.2757	292	0.3298	292	0.3369
	3	498	619	0.7495	498	0.3276	498	0.3350
	4	414	15733	17.9227	414	0.3291	414	0.3364
	5	469	1958	2.2977	469	0.3263	469	0.3383
		No. of opt.			5		5	
11	1	539	11614	13.3561	539	0.3645	539	0.3538
	2	676	14469	16.4799	676	0.3488	676	0.3397
	3	845	10094	11.6457	845	0.3408	845	0.3644
	4	429	1344	1.5238	429	0.3509	429	0.3535
	5	423	23246	26.3099	423	0.3337	423	0.3476
		No. of opt.			5		5	
12	1	627	10443	11.9636	627	0.3405	627	0.3560
	2	769	7801	8.7461	769	0.3534	769	0.3849
	3	602	97941	111.6275	602	0.3294	602	0.3577
	4	630	22908	26.8492	630	0.3297	630	0.3601
	5	689	45890	53.9377	689	0.3384	689	0.3591
		No. of opt.			5		5	
13	1	718	49918	59.1163	718	0.3347	718	0.3889
	2	573	967292	1.8000e+03	573	0.3354	573	0.3583
	3	749	349384	653.6059	749	0.3298	749	0.3660
	4	828	82155	161.0544	828	0.3381	828	0.3625
	5	692	932627	1.7117e+03	692	0.3296	692	0.3621
		No. of opt.			5		5	
14	1	1054	19719	36.7386	1054	0.3465	1054	0.3839
	2	944	148623	208.8141	944	0.3386	944	0.3773

	3	881	135901	225.9353	881	0.3288	881	0.3636
	4	617	1241576	1.8000e+03	617	0.3302	617	0.3644
	5	949	500964	912.6224	949	0.3323	949	0.3743
		No. of opt.			5		5	
15	1	1163	999952	1.8000e+03	1163	0.3428	1163	0.4000
	2	1354	979356	1.8000e+03	1354	0.3362	1354	0.3645
	3	1045	1408198	1.8000e+03	1045	0.3317	1045	0.3662
	4	1095	1542277	1.8000e+03	1095	0.3308	1095	0.3930
	5	1194	1126274	1.8000e+03	1194	0.3435	1194	0.3881
		No. of opt.			5		5	

Note: for the results of the table (3) for the problem (SP), the exact method (BAB) guarantee a global optimum for NP-hard problem, but the time required that grows exponentially with size of the problem, and often only small or medium sized instances can be solved almost demonstrable optimality. In this case, the only possibility for large causes is to use LSAs.

Table (4): The results of (LSAs) for middle size problems (SP).

$1//(\sum C_j + \sum T_j + \sum E_j + T_{max} + E_{max})$						
N	EX	best	DM	Time	SM	Time
20	1	2192	2192	17.1704	2192	17.9940
	2	1980	1980	17.2340	1980	18.1652
	3	2307	2307	17.2693	2307	17.8001
	4	1614	1614	17.0662	1614	17.9575
	5	1480	1480	17.2500	1480	18.1025
		No. of best	5		5	
25	1	3074	3074	17.4439	3074	18.0000
	2	3576	3576	17.3106	3576	18.0559
	3	3304	3304	17.2883	3304	17.9902
	4	3252	3252	17.3190	3252	17.9990
	5	2868	2868	17.0024	2868	19.0635
		No. of best	5		5	
30	1	4617	4617	17.7994	4617	19.7571
	2	4326	4326	17.2849	4326	19.3264
	3	4484	4484	17.5013	4484	19.2168
	4	4381	4381	17.6284	4381	19.4044
	5	4144	4144	17.4485	4144	19.1458
		No. of best	5		5	
40	1	8391	8391	17.8830	8391	18.3505
	2	8229	8229	18.5320	8229	18.5105

 https://doi.org/ 10.30526/2017.IHSCICONF.1817

	3	10128	10128	17.7782	10128	18.4321
	4	7423	7423	18.5639	7423	18.4112
	5	6570	6571	18.3700	6570	18.5121
		No. of best	4		5	
50	1	12294	12294	17.8408	12294	18.6126
	2	12235	12235	18.3901	12235	18.7420
	3	12943	12943	18.1681	12943	18.7678
	4	9427	9427	17.9928	9427	18.6223
	5	11453	11453	17.9824	11453	18.7382
		No. of best	5		5	
75	1	26660	26660	18.1376	26660	19.1078
	2	24708	24708	18.8023	24708	19.3242
	3	25066	25066	18.8975	25066	19.2128
	4	24067	24067	18.9234	24067	19.1542
	5	23517	23517	18.2172	23517	19.2733
		No. of best	5		5	
100	1	47533	47533	19.5215	47533	20.2826
	2	44798	44798	19.2866	44798	20.2693
	3	51191	51191	19.5268	51191	20.1367
	4	45303	45303	18.9843	45303	20.4310
	5	42456	42456	19.8174	42456	20.1911
		No. of best	5		5	

Table (5): The results of (LSAs) for large size problems (SP).

$1//(\sum C_j + \sum T_j + \sum E_j + T_{max} + E_{max})$						
N	EX	Best	DM	Time	SM	Time
150	1	114782	114782	21.3637	114782	21.8264
	2	105271	105271	20.8970	105272	21.1729
	3	113900	113900	20.9309	113900	22.3416
	4	107808	107808	21.0479	107808	22.1926
	5	105044	105044	20.9315	105044	21.8763
		No. of best	5		4	
200	1	190272	190272	23.3108	190272	27.7667
	2	195197	195197	22.6171	195197	24.3901
	3	201662	201662	22.7381	201662	23.2338
	4	165780	165780	22.5827	165780	23.1716
	5	174895	174895	22.6055	174896	23.5996
		No. of best	5		4	
500	1	1249270	1249270	32.5923	1249270	34.1950

	2	1221710	1221710	34.1415	1221710	34.0155
	3	1221949	1221949	33.5762	1221950	33.7248
	4	1128689	1128689	33.3125	1128691	33.6118
	5	1082602	1082602	33.2352	1082603	33.2889
		No. of best	5		2	
1000	1	5192233	5192233	50.9866	5192235	51.6225
	2	4985474	4985474	50.0319	4985474	51.0978
	3	4938789	4938789	49.7726	4938789	51.2369
	4	4569662	4569662	49.7137	4569662	50.9907
	5	4212750	4212750	49.6992	4212751	51.4793
		No. of best	5		3	
2000	1	19815089	19808339	86.6824	19808339	89.8018
	2	19527160	19527160	87.8703	19527160	88.2342
	3	19826853	19826853	87.6722	19826853	90.6310
	4	17448482	17448482	88.4805	17448482	88.4856
	5	17743289	17743289	87.6191	17743289	88.2204
		No. of best	5		5	
2500	1	3124028	3124028	112.0557	31224028	114.6281
	2	30408720	30408720	108.1435	30408722	111.4571
	3	30655008	30655008	108.8118	30655008	109.5242
	4	27788565	27788565	107.5072	27788565	110.2695
	5	27166817	27166817	111.9192	27166817	112.1854
		No. of best	5		4	
3000	1	44450378	44450378	127.9788	44450378	129.7762
	2	44309450	44309450	128.7811	44309450	128.7734
	3	45225080	45225080	126.3454	45225080	127.8855
	4	39495210	39495210	126.7486	39495210	128.4215
	5	40045879	40045879	127.8283	40045879	127.8793
		No. of best	5		5	
4000	1	79877216	79877216	164.5052	79877216	164.7037
	2	79609210	79609210	168.6264	79609210	165.2966
	3	78861024	78861024	164.0467	78861024	164.4531
	4	71495346	71495346	175.1210	71495346	166.4173
	5	70029765	70029765	165.5078	70029765	166.7640
		No. of best	5		5	
5000	1	123462832	123462832	207.2337	123462832	214.6064
	2	121049540	121049540	214.2919	121049540	216.2722
	3	110342921	110342921	206.2056	110342921	223.8944
	4	109574701	109574701	225.0643	109574701	204.9242
	5	111141203	111141203	204.3804	111141203	206.6917
		No. of best	5		5	

4. Conclusion

In this thesis, the problems of scheduling jobs on one machine for a variety of multicriteria are considered.

We propose algorithm, which gave set of efficient solutions for the problem (P) $1//(\sum C_i, \sum T_i, \sum E_i, T_{max}, E_{max})$. A local search algorithm (DM and SM) are used to find near optimal solution for problem (SP) of size (5000) jobs.

References

[1] T.S. Abdul-Razaq and Z.M. Ali "Minimizing the Total Completion Time, the Total Tardiness and the Maximum Tardiness", Ibn Al-Haitham. J. for pure & Appl. Sci. 28, 155-170. 2015

[2] T.S. Abdul-Razaq and Z.M. Ali "Minimizing the Total Completion Time, the Total Earliness and the Maximum Earliness".2016

[3] T.S. Abdul-Razaq, Manal G. Ahmed and H. Manal . Ibrahim, " Equivalent between Weighted Earliness and Weighted Tardiness Problems On A single Machine", Al-Mustansiriyah J. Sci. 2013

[4] T.S Abdul-Razaq and A.O. Akram . "Minimizing multi-criteria on a Single machine" To appear in Journal of Education Al- Mustansiriyah university. .2017

[5] S.S. Al-Assaf. "Solving multiple objectives scheduling problems", M.Sc. thesis Univ. of Al- Mustansiriyah, College of Science, Dep. of Mathematics (2007).

[6] Du, J. Leung, J. Y. T. Minimizing total tardiness on one processor is NP-hard, Mathematics of Operations Research, 15, 483-495. 1990

[7] S. French . "sequencing and scheduling : An introduction to the mathematical of job shop " ,john wiley and sons, New York, (1982).

[8] J.A. Hoogeveen, S.L. van de Velde . " Minimizing total completion time and maximum cost simultaneously is solvable in polynomial time", Operations Research Letters 17, 205–208(1995).

[9] J.R. Jackson. " Scheduling a production line to minimize maximum tardiness", Research Report 43, Management Sciences Research Project, UCLA(1955).

[10] 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 Conference, 803-809. 2005

[11] E. Lawler . "A pseudo-polynomial algorithm for sequencing jobs to minimize total tardiness" Annals of Research Letters, 1:207–208, 1977.

[12] G.Moslehi, R.Moghaddam, M.Vasei, and A.Azaron . "Optimal scheduling for a single machine to minimize the sum of maximum earliness and tardiness considering idle insert ",Applied Mathematics and Computation 167 , 1430–1450(2005).

[13] Smith W.E. , "Various optimizers for single stage production", Naval Research Logistics Quarterly 3/1, 59-66(1956).

[14] E. Steure Ralph. "Multiple Criteria Optimization: Theory, Computation and Application" willey, (1986).

[15] R. Tadei, A. Grosso, , and F. Della Croce . "Finding the pareto-optima for the total and maximum tardiness single machine problem", Discrete Applied Mathematics 124, 117-126. 2002

[16] L.N. Van Wassenhove and F. Gelders "Solving a bicriterion scheduling problem", European Journal of Operational Research 4/1, 42-48(1980).