Journal of AL-Qadisiyah for computer science and mathematics
Vol.6 No.1 Year 2014

Page 57-74

Local Search Methods to Solve Multiple Objective Function

Al-Zuwaini Mohammad Kadhim
Math. Dep. College of Computer Science and Mathematics
Thi-qar University
mkzz50@yahoo.com

Najah Ali Husein
Math. Dep. College of Education
Al-Qadisiyh University
najah_math2006@yahoo.com

Recived :17\9\2012 Recived : 17\2\2013 Accepted :6\3\2014

طرائق البحث المحلية لحل دالة هدف متعددة

محمد كاظم الزويني
قسم الرياضيات
كلية علوم الحاسبات والرياضيات
جامعة ذي قار

نجاح علي حسين
قسم الرياضيات
كلية التربية
جامعة القادسية

المستخلص:

تناولنا في هذا البحث دراسة مسألة جدولة n من النتاجات (Jobs) على ماكنة واحدة. هدفنا في هذه الدراسة هو ايجاد الحلول التقريبية (Near optimal solutions) لجدولة n من النتاجات لتصغير دالة الهدف وهي الكلفة الكلية لزمن انسياب النتاجات وكلفة أكبر تبكير عندما يكون للنتاجات أزمنة تحضير غير متساوية. حيث قمنا بتطوير ومقارنة واختبار بعض طرائق البحث المحلية:
(الطريقة التنازلية, محاكاة الصب, طريقة إبدال الأزواج المتجاورة, الخوارزمية الجينية) للمسألة وتحرينا عن تأثير تغاير المعلمات لهذه الطرائق. وتحليل حلولها الأولية حسابياً، عملياً ومن خلال الخبرة الحسابية وجد، بأن خوارزميات البحث المحلي تستطيع حل المسألة إلى (23000) نتاج بوقت معقول، كذلك وجدنا إن (الخوارزمية الجينية) هي الأفضل للمسألة عندما يكون الحجم اقل أو مساوي لـ(1500) نتاج، أما للمسائل من حجم اكبر كانت طريقة محاكاة الصب هي الأفضل.

Abstract

In this paper we considered the problem of scheduling n jobs on a single machine. Our aim in this study is to find the near optimal solution to minimize the cost of total flow time and maximum earliness with unequal ready times.

Different local search methods: (Descent Method, Adjacent Pairwise Interchange Method, Simulated Annealing, Genetic Algorithm) are developed, compared, and tested for the problem. We investigate the influence of the parameters variance for these local search methods, and empirically analyze their starting solutions. Computational experience found that these local search algorithms can solve the problem up to (23000) jobs with reasonable time. Also we found that: the Genetic algorithm is the best local search heuristic algorithm for our problem when the size is less than or equal to (1500) jobs, and for problems of large size the Simulated Annealing was recommended.

Keywords: Flow time; Maximum earliness; Scheduling; Ready time.

Mathematics Subject Classification : 90C47

1. Introduction

The problem of sequencing n jobs on one machine under different assumptions and multiple criteria are considered extensively. In this study the objective function to be minimized consists of two criteria with unequal ready times: sum of flow time denoted by $\sum F_i$ plus maximum earliness denoted by E_{max} . We assume that the two criteria have the same importance. Denote this problem by $1/r_i/\sum F_i + E_{max}$.

This problem is of a remarkable importance in addition to processing and minimizing the time of the flow of works on the machine. This is achieved from the time of the arrival in the work site (when it is ready for working on the machine) to the time of the work achievement. Furthermore it is possible to reduce the storage time for the works which require from the achievement till delivery to the beneficiaries. The process of storage is sometimes expensive and complex. The processing of such type of problems has considerable importance especially in the field of agriculture and industry. This is especially true when handing the problems of factories which produce items with short periods of validity for use such as food, chemical substance, serums, crops and fruits.

The following are some of Literature Review:

Koksalan et al (1998) [10] proposed a heuristic to "generate all approximately efficient sequences " for the problem to minimize the flow time and maximum earliness on a single machine . Ahmet and koksalan (2003)[2], used Genetic algorithm to solve the scheduling problem of the total completion times and the maximum earliness. Kurz and conterbury (2005) [11] used genetic algorithm to find the set of efficient point for $1//(\sum C_i, E_{\max})$ problem. Al-Assaf (2007) [3] used the BAB algorithm to find the optimal solution for the problem $1//\sum C_i + E_{\max}$ and proposed a polynomial algorithm with in special range for the problem $1//(\sum C_i, E_{\max})$.

Huang and Yang (2009) [8] presents an algorithm for efficient scheduling in terms of total flow time and maximum earliness.

Al-Zuwaini and Husein, N. A. (2012)[4] used efficient branch and bound technique with effective upper bound and valid lower bound for the problem $1/r_i/\sum F_i + E_{\max}$, also they proved special cases and dominance rules for this problem.

2. Sequence Rules for Machine Scheduling Problems

1) SPT: Jobs are sequenced in non – decreasing order of processing times, (this rule is well known to minimize $\sum C_i$) for $1//\sum C_i$ problem. [13]

2) SRT: Jobs are sequenced in non – decreasing order of release dates , (this rule is well known to minimize C_{max}) for $1/r_i/C_{max}$ problem.[6]

3) MST: Jobs are sequenced in non – decreasing order of their slack times $S_i = d_i - p_i$, (this rule is well known to minimize E_{max}) for $1//E_{max}$ problem. [7]

3. Formulation of the Problem

The general problem of scheduling jobs on a single machine to minimize the total cost can be stated as follows: A set of n independed jobs N={1,2,….,n} which has to be scheduled without preemption on a single machine that can handle at most one job at a time. The machine is assumed to be continuously available from time zero onwards and no precedence relationship exists between jobs. Each job j, $j \in N$ has an integer processing time P_j, a release date r_j and ideally should be completed at its due date d_j. For any given schedule (1,2,…,n), the flow time of job j, F_j and the maximum earliness E_{max} can be respectively defined as:

$F_j=C_j-r_j$, where C_j be a completion time for job j, given by the relationship:
$C_1 = r_1 + p_1$, $C_j = \max\{r_j, C_{j-1}\} + p_j$ for j=2,3,…,n
and $E_{\max} = \max_{1\le j\le n}\{E_j\}$, $E_j = \max\{d_j - C_j, 0\}$, j=1,2,…….,n.

The objective is to find the schedule that minimize the sum of the total flow time and maximum earliness costs of all jobs with release dates on a single machine (i.e. minimize the multiple objective function (MOF) denoted by $\left(\sum_{j=1}^{n} F_j + E_{\max}\right)$. It is clear that our model differs from the other models (See for example).

Koksalan et al. (1998) [10], Ahmet and Koksalan (2003) [2], Kurz and Canterbury (2005) [11], AL-Assaf (2007) [3], Huang and Yang (2009) [8]. Here we consider a more general and realistic problem dealing with arbitrary release dates. The problem is strongly NP-hard because the $1/\ /\sum C_i + E_{\max}$ problem with zero release date is NP-hard [10][2][3].

Our scheduling problem can be state mathematically more precisely as follows:

Given a schedule δ = (1,2,….,n), then for each job $j \in \delta$ the flow time F_j and the maximum earliness $E_{\max}$ can be calculated. The objective is to find a schedule, $\sigma = (\sigma(1), \sigma(2), \ldots, \sigma(n))$ belong to a neighborhood of δ that minimize the total cost $Z(\sigma)$, where

$$Z(\sigma) = \sum_{j=1}^{n} F_{\sigma(j)} + E_{\max}(\sigma).$$

Let S be a set of all schedules, $|S| = n!$, then we can formulate our problem in mathematical form as:

$$\left.\begin{array}{ll}
M = \min_{\sigma\in S}\{Z(\sigma)\} = \min_{\sigma\in S}\left\{\sum_{j=1}^{n} F_{\sigma(j)} + E_{\max}(\sigma)\right\} & \\
S.to: & \\
C_{\sigma(j)} \ge r_{\sigma(j)} + p_{\sigma(j)} & j=1,2,\ldots,n \\
C_{\sigma(j)} \ge C_{\sigma(j-1)} + p_{\sigma(j)} & j=2,\ldots\ldots,n \\
F_{\sigma(j)} = C_{\sigma(j)} - r_{\sigma(j)} & j=1,2,\ldots,n \\
E_{\sigma(j)} \ge d_{\sigma(j)} - C_{\sigma(j)} & j=1,2,\ldots,n \\
p_{\sigma(j)} > 0,\ r_{\sigma(j)} \ge 0 & j=1,2,\ldots,n \\
E_{\sigma(j)} \ge 0,\ F_{\sigma(j)} \ge p_{\sigma(j)} & j=1,2,\ldots,n
\end{array}\right\} \quad (P)$$

mation Solution of

Let t be a time at which a machine is available after it ;

$R_i(t) = \max(t, r_i)$ the earliest beginning time of job i at time t.

$C_i(t) = R_i(t) + P_i$ the earliest completion time of job i at time t.

$G(i,t) = R_i(t) + C_i(t)$ priority rule for total flow time of job i at time t.

Then, given a set of jobs N={1,2,…,n}

Step (1) : Initialized t = 0 , A = {1,2,….,n} and $\sigma = \phi$

Step (2) : Select job i with $\min_{i \in A}$ G(i,t). Break ties by choosing i with $\min\{R_i(t)\}$, and further ties by choosing i with min d_i.

Step (3) : Update t , A and σ , such that t = $C_i(t)$, A=A-{i}, $\sigma = \sigma \cup \{i\}$

Step (4) : If A $\neq \phi$, return to step 2.

Step (5) : Compute UB= $\sum_{i=1}^{n} F_i(\sigma) + E_{\max}(\sigma)$.

5. Near optimal solution by using local search methods

Obviously the problems including multiple criteria are more difficult than those with single criteria. This is the reason why it appears from the analysis of the BAB method result that often weak. So there is a need for local search methods to treat a large size instances problem. This is the main aim of the present paper. In this section different local search methods are developed, compared and tested for the problem **(P).**

5.1 Descent Method (DM) :

This method is a simple form of local search methods. It can be executed as follows :

Step (1): Initialization

The initial current solution obtained from the Construction of heuristic described in section (4) is to be the initial upper bound (UB) with its current sequence $\sigma = (\sigma(1), \sigma(2), ..., \sigma(n))$ and objective function $f(\sigma)$.

Step (2): Neighbor generation

The neighbor is swap neighbor (select two arbitrary jobs i and j (i ≠ j) not necessary be adjacent and interchange them). The neighbor $\sigma^* = (\sigma^*(1), \sigma^*(2), ..., \sigma^*(n))$. Let the objective function value of this neighbor be $f(\sigma^*)$.

Step (3): Acceptance test

In this step, we are going to test whether to accept σ^* or retain to σ for the previous neighbor as follow.

a- If $(f(\sigma^*) < f(\sigma))$, then σ^* replace σ as the current solution and we set $f(\sigma) = f(\sigma^*)$, then go to step (2) (Neighbor generation).

b- Otherwise (i.e. $f(\sigma^*) \geq f(\sigma)$), then σ retain as the current solution and we retain to step (2) (neighbor generation).

Step (4): Termination condition

After (30,000) iterations the algorithm is stopped at a near optimal solution.

5.2 Adjacent Pairwise Interchange Method (APIM)

This method defined by a pair interchange operators which interchange elements (jobs) at position (i) and (i+1) for a given sequence (i= 1, 2,……, n-1)

Now we are going to describe the steps of (APIM)

Step (1): Initialization

Is the same as initialization in DM and with its objective function value $f(\sigma)$

Step (2): Neighbor generation

In order to improve the sequence σ, the position of two adjacent jobs $\sigma(i)$, $\sigma(i+1)$, $1 \leq i \leq n-1$ are transposed. Hence a new sequence σ^* is obtained with its objective function $f(\sigma^*)$

Step (3): Acceptance test

If the improvement is made [i.e. $f(\sigma^*) < f(\sigma)$], then the two jobs are left in their new position. On the other hand, the two jobs are replaced in their original positions. The procedure is then repeated from step(2) and other possibilities are considered in a similar way.

Step (4): Termination condition :

After (30,000) iterations the algorithm is stopping at a near optimal solution.

5.3 Simulated Annealing (SA):

In this method improving and neutral moves are always accepted. While deteriorating moves are accepted according to a given probability acceptance function[12].

The following steps describe SA.

Step (1) : Initialization

Is the same as initialization in DM and with its objective function value $f(\sigma)$.

Step (2) : Neighborhood generation

The neighbor σ^* of the current solution σ is swap neighbor and compute its objective function value $f(\sigma^*)$.

Step (3) : Acceptance test

The initial temperature is 10^0 and $T^{new} = hT^{old}$ where $0<h<1$ (h is chosen arbitrary) (h=0.9), then we compare between $f(\sigma)$ and $f(\sigma^*)$ as follows:

a- If $f(\sigma^*) \le f(\sigma)$, then σ^* is accepted and replaced σ as the current solution.

b- If $f(\sigma^*) > f(\sigma)$, and $e^{-\Delta/T} >$ R, $\Delta = f(\sigma^*) - f(\sigma)$
where 0<R<1, R is chosen arbitrary. Then σ^* is accepted and replace σ as the current solution, else we reject σ^* and retain to σ.

Step (4): Termination condition :

After (30,000) iterations the algorithm is stopping at a near optimal solution.

5.4 Genetic Algorithm (GA)

Genetic algorithms are global search and optimization techniques modeled from natural genetics. They date back to the early work described by John Holland**.** It works on a randomly generated candidate solution pool, which is usually called "population". Each encoded candidate solution is called "chromosome". During the searching process, the selection, crossover and mutation operators are executed repeatedly until the stop criteria is satisfied[15]. In the following we describe each of the mechanism for our scheduling problem briefly :

1. Initialization

The initial population can be generated at random or can be constructed by using heuristic methods. In this paper we start with m = 120, 113 from them generated randomly and the remaining seven are given by SPT rule, MST rule, SRT rule, the construction heuristic which is used in section (4), order the jobs according to non – decreasing order of d_i - (r_i + p_i), DM which is used in subsection (5.1) with termination condition (after 1000 iterations) and SA which is used in subsection (5.3) with termination condition (after 1000 iterations).

2. New population

A new population is created by repeating the following substeps until the new population is completed.

a. Selection :

Selecting the individuals according to fitness value that will usually form the next generating's parents.

b. Crossover :

Crossover is the breeding of two parents to produce a single child. The child has features from both parents and thus may be better or worse than either parent according to the objective function. Homogeneous mixture crossover (HMX) [1] are applied on each pair of parent solutions to generate two new solutions (children).

c. Mutation

Pairwise (swap) mutation is applied on each pair of parent solutions to generate two new solutions (children).

3. Termination Condition:

The GA procedure stops when a fixed number of generations (or iterations) are executed here (200) iterations. This means that the GA procedure continues until the population is converged to a good, if not optimal solution to our problem (P).

6. Computational Results of Local Search Algorithms and Comparison

6.1 Test Problems

There exists in the literature a classical way to randomly generate test problems of scheduling problems.

- The processing time P_i is uniformly distributed in the interval [1,10].
- The release date r_i is uniformly distributed in the interval [0, α P], where [α = 0.125, 0.25, 0.50, 0.75, 1.00] and $P=\sum_{i=1}^{n} P_i$.
- The due date d_i is uniformly distributed in the interval [P(1-TF-RDD/2),P(1-TF+RDD/2)];where $P=\sum_{i=1}^{n} P_i$ depending on the relative range of due date (RDD) and on the average tardiness factor (TF).

For both parameters, the values 0.2, 0.4, 0.6, 0.8 and 1.0 are considered. For each selected value of n where n is the number of jobs, ten problems were generated.

6.2 Computational Results

All local search algorithms in this paper (Decent Method, Adjacent Pairwise Interchange Method, Simulated Annealing, Genetic Algorithm), are coded in Matlab 7.9.0 (R2009b) and implemented on Intel (R) core (TM) i3 CPU M380 @ 2.53 GH2, with RAM 4.00 GB personal computer. In our computational, we use the condition that: if the solution of an example with " n " jobs for any algorithm is not appear after (600) seconds i.e. (10 minutes) from its run; then this example is unsolved and this algorithm is active until the problem of size " n" . These criteria were used by Stoppler and Bierwrith [14].

6.2.1 Comparative Effective of Local Search Algorithms

Table (1) shows for each algorithm, the value of objective function and how many it can catch the optimal value for each value of " n " (problem size). In addition, describes the deviation of local search methods from the optimal solution. The optimal solution for examples in table (1) was found by using BAB algorithm in [4].

Table (2) shows the values of each local search algorithms and how many time that each of them catch the best value, where:

Optimal= the optimal value which is obtained by using BAB method.

SM = the value found by Simulated annealing.

DM = the value found by decent method.

APIM = the value found by adjacent pairwise interchange method

GA = the value found by Genetic algorithm.

No of opt.= number of examples that catch the optimal value.

Av. Time = the average of time for (10) examples for each algorithm.

Best = the best value.

No. of best = number of examples that catch the best value.

- = refer to the unsolved example.

Table (1) : The performance of local search methods and the optimal solution for n $\in$ {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}

n	EX	Optimal	SA	DM	APIM	GA
5	1	71	71	71	71	71
	2	64	64	64	64	64
	3	90	90	90	90	90
	4	31	31	31	31	31
	5	35	35	35	35	35
	6	46	46	46	46	46
	7	62	62	62	62	62
	8	78	78	78	78	78
	9	71	71	71	71	71
	10	77	77	77	77	77
No of opt.			10	10	10	10
Av. Time			0.4723	0.4443	0.4391	0.5335

n	EX	Optimal	SA	DM	APIM	GA
10	1	299	299	299	299	299
	2	185	185	185	192	185
	3	262	262	262	266	262
	4	177	177	177	180	177
	5	208	209	209	212	209
	6	189	189	189	189	189
	7	167	167	167	170	167
	8	219	219	219	219	219
	9	218	218	218	218	218
	10	267	267	267	269	267
No of opt.			9	9	4	9
Av. Time			0.5069	0.4702	0.4759	0.6286
15	1	677	677	678	684	677
	2	392	392	392	392	392
	3	616	616	616	628	616
	4	416	416	416	419	416
	5	474	474	474	474	474
	6	545	545	545	557	545
	7	419	419	419	419	419
	8	542	542	542	544	542
	9	495	495	495	495	495
	10	465	465	465	465	465
No of opt.			10	9	5	10
Av. Time			0.5249	0.4921	0.4989	0.6923
20	1	807	807	807	807	807
	2	697	697	698	698	697
	3	906	906	907	918	906
	4	814	815	815	815	815
	5	829	829	829	833	829
	6	1043	1043	1043	1043	1043
	7	708	708	708	708	708
	8	551	551	552	554	551
	9	764	764	764	764	764
	10	681	681	681	681	681
No of opt.			9	6	5	9
Av. Time			0.5590	0.5275	0.5220	0.7842
25	1	1294	1294	1294	1294	1294
	2	1225	1225	1225	1243	1225
	3	1422	1422	1422	1422	1422
	4	989	989	989	999	989
	5	1306	1317	1317	1321	1317
	6	1468	1468	1468	1468	1468
	7	1363	1363	1363	1363	1363
	8	1060	1060	1060	1063	1060
	9	933	933	933	933	933
	10	1053	1063	1063	1063	1063
No of opt.			8	8	5	8
Av. Time			0.6268	0.5843	0.5698	0.8925

n	EX	Optimal	SA	DM	APIM	GA
30	1	1496	1496	1503	1500	1496
	2	1850	1868	1868	1868	1850
	3	1881	1881	1881	1881	1881
	4	1584	1622	1622	1622	1584
	5	1319	1319	1319	1320	1319
	6	1871	1871	1871	1871	1871
	7	1566	1566	1567	1567	1566
	8	1890	1893	1893	1893	1893
	9	1740	1740	1740	1740	1740
	10	1469	1469	1470	1470	1469
No of opt.			7	4	3	9
Av. Time			0.6043	0.5728	0.5729	1.0076
35	1	1873	1873	1876	1911	1873
	2	2230	2230	2230	2230	2230
	3	1931	1939	1939	1984	1931
	4	1761	1761	1775	1793	1761
	5	2028	2029	2031	2033	2028
	6	1835	1835	1835	1835	1835
	7	2363	2363	2369	2389	2363
	8	2115	2128	2128	2128	2128
	9	2541	2542	2542	2566	2541
	10	2079	2079	2079	2093	2079
No of opt.			6	3	2	9
Av. time			0.6995	0.6602	0.6429	0.1399
40	1	2724	2724	2725	2747	2724
	2	2980	2981	2980	3011	2980
	3	2823	2823	2823	2824	2823
	4	2868	2868	2868	2876	2868
	5	2469	2469	2469	2469	2469
	6	2649	2649	2672	2672	2649
	7	2649	2660	2655	2685	2655
	8	2018	2018	2018	2022	2019
	9	2692	2692	2692	2695	2692
	10	2323	2323	2325	2333	2323
No of opt.			8	6	1	8
Av. time			0.6763	0.6470	0.6412	1.2783
45	1	4555	4580	4588	4598	4555
	2	3881	3932	3896	3953	3892
	3	4103	4122	4122	4130	4122
	4	3980	3981	3981	3981	3981
	5	3616	3625	3625	3625	3625
	6	3411	3411	3411	3411	3411
	7	3576	3578	3615	3615	3578
	8	3917	3917	3917	3917	3917
	9	3594	3594	3594	3594	3594
	10	4302	4302	4303	4315	4302
No of opt.			4	3	3	5
Av. time			0.7192	0.7161	0.7042	1.4078

n	EX	Optimal	SA	DM	APIM	GA
50	1	3973	3973	3974	3984	3981
	2	5029	5101	5101	5105	5029
	3	3837	3837	3837	3837	3837
	4	3979	4024	4024	4024	4024
	5	4590	4590	4590	4590	4590
	6	4175	4215	4177	4215	4177
	7	4886	4886	4908	4899	4899
	8	4710	4713	4713	4713	4713
	9	3605	3605	3605	3605	3605
	10	3839	3842	3841	3881	3841
No of opt.			5	3	3	4
Av. time			0.7235	0.6993	0.6869	1.5654

Table (2): The performance of local search methods and the best solution for

$n \in \{75, 100, 500, 1000, 1500, 2000, 5000, 10000, 15000, 23000\}$

n	EX	Best	SA	DM	APIM	GA
75	1	9860	9861	9861	9861	9860
	2	11158	11158	11172	11255	11174
	3	9797	9797	9798	9798	9797
	4	10799	10816	10799	10855	10816
	5	8688	8688	8688	8688	8697
	6	9598	9611	9618	9647	9598
	7	10289	10289	10289	10289	10289
	8	9962	9962	9962	9967	9963
	9	9910	9910	9910	9910	9910
	10	9879	9879	9883	9899	9884
No of best.			7	5	3	5
Av. time			0.8702	0.8384	0.8491	2.4570
100	1	17168	17168	17168	17168	17168
	2	17953	17953	17953	17953	17953
	3	16746	16757	16756	16841	16746
	4	14828	14828	14828	14828	14829
	5	16958	16958	16960	16965	16964
	6	18808	18824	18824	18876	18808
	7	16756	16757	16756	16756	16757
	8	19565	19581	19587	19740	19565
	9	15538	15538	15538	15539	15544
	10	17535	17535	17535	17537	17535
No of best.			6	6	4	6
Av. Time			0.9872	0.9655	0.9687	3.5881

n	EX	Best	SA	DM	APIM	GA
500	1	436361	436422	436407	436361	436463
	2	404995	404995	405141	405794	405019
	3	415284	415490	415371	415564	415284
	4	454259	454363	454279	454545	454259
	5	390066	390077	390078	390066	390078
	6	396251	396265	396262	396251	396278
	7	416436	416439	416442	416436	416442
	8	412551	413031	413115	414130	412551
	9	388906	389075	388958	389187	388906
	10	413173	413173	413343	413817	413375
No of best.			2	0	4	4
Av. time			3.4626	3.4001	3.39859	54.8531
1000	1	1687780	1687962	1687986	1687780	1687911
	2	1604536	1604536	1604790	1605075	1604648
	3	1623290	1623351	1623356	1623290	1623358
	4	1641607	1641607	1642341	1643534	1642121
	5	1564666	1564668	1564885	1565164	1564666
	6	1680240	1680451	1680240	1680686	1680731
	7	1514604	1514661	1514850	1516227	1514604
	8	1576885	1577112	1576885	1577496	1576932
	9	1603403	1603575	1603575	1603403	1603575
	10	1593253	1593253	1593704	1594219	1593476
No of best.			3	2	3	2
Av. time			6.9457	6.9031	6.9008	206.5347
1500	1	3612385	3612385	3612487	3613122	3612385
	2	3664685	3665130	3664685	3665427	3664785
	3	3600883	3601626	3601634	3600883	3601542
	4	3559265	3559414	3559387	3560146	3559265
	5	3722875	3723432	3723463	3722875	3723470
	6	3639492	3641967	3640435	3646089	3639492
	7	3721909	3722161	3722145	3721909	3722168
	8	3568594	3569133	3569135	3569218	3568594
	9	3683032	3684318	3683032	3686314	3683350
	10	3672782	3672782	3673739	3675435	3673189
No of best.			2	2	3	4
Av. time.			10.9217	10.8710	10.8766	455.3346
2000	1	6549334	6549334	6551481	6553498	•
	2	6312224	6312224	6312441	6312906	•
	3	6463538	6463870	6463538	6464299	•
	4	6464788	6465710	6464788	6469977	•
	5	6329095	6329095	6329531	6329229	•
	6	6675689	6675689	6677398	6678504	•
	7	6504549	6504804	6504549	6508568	•
	8	6536180	6536180	6537120	6538374	•
	9	6641705	6641705	6642284	6646912	•
	10	6278827	6279260	6279261	6278827	•
No of best.			6	3	1	
Av. time			15.3756	15.3329	15.3703	

n	EX	Best	SA	DM	APIM	GA
5000	1	40456994	40459184	40459208	40456994	•
	2	40150529	40152258	40152247	40150529	•
	3	40274315	40274315	40276206	40276908	•
	4	40135864	40138108	40138069	40135864	•
	5	40090251	40090251	40091780	40109045	•
	6	39056811	39058739	39058762	39056811	•
	7	39914461	39918838	39914461	39919352	•
	8	40133542	40138055	40133542	40137511	•
	9	39912837	39912837	39915131	39913863	•
	10	39863164	39864321	39864312	39863164	•
No of best.			3	2	5	
Av. time			49.1192	49.1201	49.1119	
10000	1	160424847	160424847	160433228	160436250	•
	2	157742822	157746896	157746882	157742822	•
	3	160334389	160338332	160338356	160334389	•
	4	162284752	162285754	162284752	162345107	•
	5	163893523	163897834	163897860	163893523	•
	6	162425732	162425732	162447895	162452997	•
	7	159465569	159468493	159465569	159469411	•
	8	160375747	160375747	160377125	160388881	•
	9	158802618	158809405	158802618	158813912	•
	10	161398212	161398212	161403956	161442366	•
No of best.			4	3	3	
Av. time			136.7695	136.7910	136.7851	
15000	1	361653559	361656620	361653559	361654152	•
	2	359028403	359028403	359042670	359045323	•
	3	362767212	362781940	362767212	362779882	•
	4	363877330	363877330	363877533	363881268	•
	5	361699572	361699572	361717418	361713096	•
	6	362547646	362552050	362552039	362547646	•
	7	364214765	364219164	364219148	364214765	•
	8	357093876	357093876	357095978	357114961	•
	9	359352988	359357166	359357170	359352988	•
	10	358007164	358011751	358011763	358007164	•
No of best.			4	2	4	
Av. time			268.4312	268.2156	282.9323	
23000	1	849474778	849476039	849474778	849553321	•
	2	850885118	850889290	850889296	850885118	•
	3	849841209	849855569	849841209	849857974	•
	4	858627492	858632416	858632394	858627492	•
	5	841890559	841895149	841895148	841890559	•
	6	854642112	854642112	854643022	854774342	•
	7	854999228	855003799	855003813	854999228	•
	8	852511194	852515848	852515876	852511194	•
	9	849715381	849718422	849715381	849725821	•
	10	846655790	846660381	846660398	846655790	•
No of best.			1	3	6	
Av. time			578.6513	574.7921	573.1042	

6.2.2 Summary of Experimental Evaluation of Local Search Methods

The computational times of all algorithms for the ($1/r_i/\sum F_i + E_{max}$) problem, with our modifications on these algorithms, are approximately the same (except for the Genetic algorithm), since the computational time of (GA) is very large as compared with the computational time of DM, APIM and SA. Indeed this difference of times comes from the way that uses to generate the new sequence in each method.

- In the following table (3), we summarize the results of table (1) by viewing how many the algorithm catch the optimal value only, and their sum, for each number of jobs and for all local search methods.

Table (3): summary of results of table (1)

n	SA	DM	APIM	GA
5	10	10	10	10
10	9	9	4	9
15	10	9	5	10
20	9	6	5	9
25	8	8	5	8
30	7	4	3	9
35	6	3	2	9
40	8	6	1	8
45	4	3	3	5
50	5	3	3	4
Sum	76/100	61/100	41/100	81/100

- In the following table (4), we summarize the results of table (2) by viewing how many the algorithm catch the best value only, and their sum.

Table (4): summary of results of table (2)

n	SA	DM	APIM	GA
75	7	5	3	5
100	6	6	4	6
500	2	0	4	4
1000	3	2	3	2
1500	2	2	3	4
2000	6	3	1	•
5000	3	2	5	•
10000	4	3	3	•
15000	4	2	4	•
23000	1	3	6	•
Sum	38/100	28/100	36/100	21/50

- In the following table (5), we give the activity of local search algorithms, (i. e. give the maximum number of jobs " n " that the local search algorithms can solve the ($1/r_i/\sum F_i + E_{max}$) problem with reasonable time, (i. e. according to the condition

that had been given in subsection (6.2).

Table (5): shows activity of the local search methods

Algorithm	Active until (maximum no. of jobs)
SA	23000
DM	23000
APIM	23000
GA	1500

7. Conclusion

In this paper, we have developed near optimal solution approaches for the one machine scheduling problem to minimize a multiple objective function for the $1/r_i/\sum F_i + E_{\max}$ problem, this problem is considered to be strongly NP-hard. The main conclusion to be drawn from our computation results is that: some of the local search heuristic algorithms can solve ($1/r_i/\sum F_i + E_{\max}$) problem of size (23000) jobs in reasonable time. Also we found that the Genetic algorithm is the best algorithm for the ($1/r_i/\sum F_i + E_{\max}$) problems of size less than or equal to (1500) jobs. And for the problems of large size the simulated annealing is more effective method for our problem.

References

[1] Abbas, I.T., "The performance of multicriteria scheduling in one machine", M.Sc. thesis Univ. of Al-Mustansiriyah, College of Science, Dep. of Mathematics (2009).

[2] Ahmet Burak, Koksalan M., "Using Genetic Algorithm for Single-Machine Bicriteria Scheduling Problems, European Journal of Operational Res,145,543-556 (2003).

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

[4] Al-Zuwaini M.K. and Husein N. A.," Branch and Bound Method to Solve Multiple Objective Function ". Journal of Thi-Qar Science, Vol.3(3) , 212-229 (2012).

[5] Arats E.H.L and Lenstra J.K.(eds.),"Local Search in Combinatorial Optimization", John Wiley and Sons, Chichest, (1997).

[6] Baker, K.R "Introduction to Sequencing and Scheduling " Wile New York, (1974).

[7] Franch, S. "Sequencing and Scheduling an Introduction to Mathematics of Job Shop", John Wiley & Sons, New York (1982).

[8] Huang R-H and Yang C-L," An algorithm for minimizing flow time and maximum earliness on a single machine", Journal of the Operational Research Society 60,873-877, (2009).

[9] Jackson J.R., "Scheduling a production line to minimize maximum tardiness" Res. Report 43 management science, Res. Project, University of California, Loss Angles, CA, (1955).

[10] Koksalan M., Azizoglu M., Kondakci S., " Minimizing flow time and maximum earliness on a single machine ", IIE Transaction 30, 192-200 (1998).

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

[12] Motaghedi-Larijani A., Haddad H. R., Laghaie K. S. and Tavakkoli-Moghaddam R., " Anew single machine scheduling problem with setup time, job deterioration and maintenance costs", International Journal of Management Science and Engineering Management,6(4):284-291, (2011).

[13] Smith W.E., "Version Optimizer for Single Stages Production", Naval Res. Logistics Quarter 3, 59-66, (1956).

[14] Stöppler S. and Brierwirth C. " The Application of parallel Genetic algorithm to the n/m/p/Cmax flow shop problem" , University of Bremen <C13,f@dhbrrz 41.bet.net>.

[15] Sun L., Cheng X. and Liang Y.," Solving job shop scheduling problem using Genetic Algorithm with penalty function", International Journal of Intelligent Information Processing,Vol.7,No.2,65-77, (2010).