Journal University of Kerbala , Vol. 16 No.1 Scientific . 2018

Solving Four Cost Multi-Objective Scheduling Problem Simultaneously

حل أربعة دوال في مسألة جدولة متعددة الأهداف سويتا

Tariq S . Abdul – Razaq [1], Hafed M . Motair[2]
[1]Dept. of Mathematics , College of Science , University of Al-Mustansiriya
[2] Dep. of Mathematics ,Open Educational College, Al- Qadisiya , Ministry of Education

Abstract

In this paper, we consider single machine scheduling problem (P) to minimize four cost functions, total completion times, total tardiness, maximum tardiness, and maximum earliness. The minimization besed on two types, in the first one we study some special cases including lexigraphical minimization of problem (P). In the second type we minimize four cost functions simultaneously and propose CTTE algorithm (total completion time, total tardiness, maximum tardiness and maximum earliness) to find the set of "non-dominated solutions" of problem (P), also improve this algorithm by using intensification procedure (IMCTTE) (Imoroved CTTE). Also we propose MOVNS (Multiobjective variable neighborhood search) algorithm based on the variable neighborhood and Intensification Procedure ideas .We compare the proposed algorithms with NSGA2 algorithm. The performance of the proposed algorithms is evaluated on a large set of test problems and the results are compared. The compu- tational results show that IMCTTE algorithm is more efficient than CTTE algorithm in both, number of "non-dominated solutions" and the controbution of "non-dominated solutions" that belong to reference set. Also we find that MOVNS algorithm give better performance than CTTE and IMCTTE algorithms for all problem instancs, and better than NSGA2 specially for small size problems .

Keywords: Multiobjective scheduling, Pareto optimal solution, Multiobjective local search algorithms

الخلاصة

تم في البحث دراسة مسأله الجدولة لماكنة واحدة (P) لتصغير أربعة دوال: مجموع زمن إتمام النتاجات ، مجموع أزمان التبكير، أكبر زمن تبكير، وأكبر زمن تأخير. تم تناول نوعين من مسائل التصغير: الأول التصغير حسب الأهمية (lexigraphical) والثاني تصغير الدوال سويتا (simultaneously). اقترحنا خوارزميه (CTTE) ثم تحسينها بخوارزمية (IMCTTE). وتم في البحث اقتراح خوارزميه البحث المحلي متعددة الأهداف (MOVNS) ومقارنه جميع الخوارزميات المقترحه مع الخوارزمية الجينية متعددة الأهداف (NSGA2). أداء الخوارزميات المقترحة تم اختباره مع مجموعه واسعه من مسائل الاختبار وبمقارنة النتائج ظهر أداء خوارزمية (IMCTTE) أفضل من خوارزمية (CTTE) وذلك حسب معياري المقارنة، وكذلك وجدنا أن أداء خوارزمية (MOVNS) أفضل من الخوارزميات المقترحة (CTTE, IMCTTE) في جميع المسائل المدروسة، وأنها أفضل من خوارزمية (NSGA2) عندما n صغيرة.

1.Introduction

Scheduling means allocating scarce production resources (machines) in order to complete tasks (jobs) under certain constraints with the aim to best fulfil specific objectives. The goal is the determination of a schedule, that specifies when and on which machine each job is to be executed [21]. This paper consider the single machine scheduling problem to find the set of "non-dominated solutions". Scheduling problems in real life applications generally involves optimization of more than criteria. These criteria often conflicting in nature and quite complex. In simple cases, the objective functions may be combined into a single objective by using a weighted sum approach, or all objectives are considered simultaneously.

The general multicriteria scheduling problem (MSP) can be formulated as follows: Minimize $F(z) = (f_1(z), f_2(z), \dots, f_S(z))$, such that $z \in Z$ where z is a sequence (solution), Z is the set of all

"feasible solutions", S is the number of objectives in the problem, $F(z)$ is the S-objective value of z and each $f_i(z), i = 1, \dots, S$ represent one minimization objective under certain constrains. An exact solution that simultaneously minimize each objective function is impossible. A reasonable solution to a multiobjective problem is to find a set of solutions called as non-domonated solutions. A "non-dominated solution" is a feasible solution $\sigma \in Z$ such that there does not exist another solution $\pi \in Z$ satisfying $f_i(\pi) \leq f_i(\sigma)$ for $i = 1,2, \dots, S$ and $f_j(\pi) < f_j(\sigma)$ for at least one objective function j otherwise σ is said to be dominated by π. If the solution is not dominated by any other solution, then it is said to be "Pareto optimal". it cannot be improved with respect to any objective without worsening at least one other objective. A "Pareto optimal" set is the set of all "non-dominated solutions" in Z, and the image of a given "Pareto optimal" set, is called the Pareto front.

Our objective is to minimize simultaneously four cost functions which are total completion time, total tardiness, maximum tardiness, and maximum earliness. According to the three field scheme this problem is denoted by $1||(\sum_j C_j, \sum_j T_j, T_{max}, E_{\max})$, this problem is NP-hard since the problem $1||\sum_j T_j$ is NP-hard [6], and any problem containing the cost function $\sum_j T_j$ as subproblem is also NP-hard [9]. Smith [18] studied the problem $1|T_{max} = 0|F(\sum_j C_j, T_{max})$ .Vanwassenhove and Gelders [20] generalized this problem to the case where $T_{max} \neq 0$, they give a set of efficient points by a peseudo polynomial algorithm. Sen and Gubta [16] studied the problem of minimizing a "linear combination" of maxmum tardiness and flow times of on a single machine and a given number of jobs. To find an optimal solution they introduce a branch and bound algorithm. Hoogveen and Van de Velde [10] find the set of "non-dominated solutions" to the $1||F(\sum_j C_j, f_{max})$ problem. Tadei, Grosso, and Della Croce [19] studied $1||(\sum_j T_j, T_{max})$ problem, they search the set of all non equivelant Pareto optima. Baker and Smith [4] introduced a mutiobjective single machine problem and, the objective is to minimize simulatanuosly total weighted completion time, maximum lateness and maximum completion time. They presented two polynomial time "dynamic programming" algorithms for the problem of minimizing a combination of total completion times or "maximum lateness" cost functions. Geiger (2004)[8], proposed multiobjective varaible neighborhood search algorithm to minimize permutation flow shop scheduling problem include different "combinations" of cost functions. Several methods are proposed for solving multiobjective optimization problems using genetic algorithm (GA) such as "vector evaluated genetic algorithm" (VEGA), which is proposed by Schaffer [15], "multiobjective genetic algorithm" (MOGA) [7], "random weighted genetic algorithm" (RWGA) [12], "non-dominated sorting genetic algorithm- 2" (NSGA2)[5](See section 4.1.1). Sioud et al.(2010) [17] using constraint programming and introduce a hybrid genetic algorithm. They considered the single machine scheduling problem with "sequence dependent setup times", the objective is to minimize total tardiness, makespan, and total earliness. Arroyo,Ottoni and Oliveria (2011)[3] applied the idea of Geiger and proposed multiobjective variable neighborhood search algorithm using intesification procedure to solve single machine scheduling problem to minimize total weighted earliness / tardiness and total flowtime criteria. Abdul-Razaq and Ibraheem (2014) [1] studied the problem $1||F(E_{max}, T_{max})$, they propose a general algorithm to find the set of approximate efficient (Pareto optimal) solution and for the problem $1||E_{max} + T_{max}$ they find the (near) optimal solution using Branch and bound (BAB) algorithm. Abdul-Razaq and Mahrooz (2015) [2] study problem including three criteria, Total Completion Times, the Total Tardiness and the Maximum Tardiness, they propose a BAB algorithm for the problem "$1||(\sum_j C_j + \sum_j T_j + T_{max})$" and find the set of "non-dominated solutions" for the problem $1||(\sum_j C_j, \sum_j T_j, T_{max})$.

The rest of the paper is organized as follows :The multiobjective problem definition is described in section 2.Section 3provides some results and special cases for problem(P), also this section include the propesed CTTE algorithm and it's improvement IMCTTE.In Section4 we provide two mutiobjective local search algorithms for single machine scheduling problems, the first is NSGA2 and the second is MOVNSS. The computational experiments to evaluate the performance of the proposed algorithms are presented in section5.Finally concluding remarks are presented in section 6.

2 Problem formulation:

Let $N = \{1,2,\dots,n\}$ be the set of jobs to be processed by a machine. p_i a processing time of a job i, d_i is a due date of a job i, for $i = 1,2,\dots,n$ .All of the jobs are available to be processed by the machine and it starts processing without interrupted, and requires p_i of time to complete its processing. Let π be a sequence of the jobs in N represented by the n-tuple $(\pi(1),\pi(2),\dots,\pi(n))$ where $\pi(i)$ is the ith job processed by the machine. The completion time of job $\pi(i)$ is given by $C_{\pi(i)} = \sum_{j=1}^{i} p_{\pi(j)}$, the tardiness of the job $\pi(i)$ is givin by $T_{\pi(i)} = \max(C_{\pi(i)} - d_{\pi(i)}, 0)$, and the earliness of the job $\pi(i)$ is givin by $E_{\pi(i)} = \max(d_{\pi(i)} - C_{\pi(i)}, 0)$. The mathematical form of the problem can be written as follows :

$$\min_{s\in S}\begin{pmatrix}\sum_j C_j\\ \sum_j T_j\\ T_{\max}\\ E_{\max}\end{pmatrix}$$

$$\left.\begin{array}{ll} S\,T & \\ C_i \geq p_i & i = 1,2,\dots,n \\ C_i = C_{(i-1)} + p_i & i = 2,3,\dots,n \\ T_i \geq C_i - d_i & i = 1,2,\dots,n \\ T_i \geq 0 & i = 1,2,\dots,n \\ E_i \geq d_i - C_i & i = 1,2,\dots,n \\ E_i \geq 0 & i = 1,2,\dots,n \end{array}\right\} \quad \dots (P)$$

Where S is the set of all schedules.

The multicriteria problems that we consider concerns simultaneous minimization of four criteria which are total completion times $\sum_j C_j$,total tardiness $\sum_j T_j$, maximum tardiness T_{max} and maximum earlines E_{max}. The total completion times a measure for average in processing inventory and the other objective functions deal with service to customers.The aim for this problem is to minimize simultaneously the cost functions $\sum_j C_j$, $\sum_j T_j$, T_{max} , and E_{max} to find the set of "non-dominated solutions" . This problem for

a givine schedule $s = (1,2,\dots,n)$ is denoted by:

$$1\,||(\textstyle\sum_j C_j(s), \sum_j T_j(s), T_{max}(s), E_{max}(s))\ \text{(P)}.$$

This problem is difficult to solve and find the exact set of all "non-dominated solutions".We propose efficient algorithms to find approximate set of "non-dominated solutions" for this problem.

3. Characterizing a non-dominated solutions for problem (P).

The following results are used to find "non-dominated solutions" for the problem (P).

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

Proof : assuming that all processing times are different .The unique SPT schedule (SPT^*) gives the absolute minimum of $\sum_j C_j$. Hence there is no schedule $\sigma \neq SPT^*$ such that :

$$\left.\begin{array}{l} \sum_j C_j(\sigma) \leq \sum_j C_j(SPT^*), \sum_j T_j(\sigma) \leq \sum_j T_j(SPT^*), \\ T_{max}(\sigma) \leq T_{max}(SPT^*)\ and\ \ E_{max}(\sigma) \leq E_{max}(SPT^*) \end{array}\right\} \dots (1)$$

with at least one strict inequality. If more than one SPT schedule exists, the SPT^* is schedule satisfying SPT rule such that jobs with equal processing times are ordered in EDD schedule. If still more than one jobs have the same processing times and same due dates, then the SPT^* is not unique, then these equal jobs are ordered in MST rule. Also any sequence do not satisfy the SPT rule can not dominate an SPT^* sequence (1), if π is an SPT order sequence but not an SPT^* sequence, it can not dominate SPT^* since :

$$\left.\begin{array}{c} \sum_j C_j(SPT^*) = \sum_j C_j(\pi)\,, \sum_j T_j(SPT^*) \le \sum_j T_j(\pi), \\ T_{max}(SPT^*) \le T_{max}(\pi)\ and\ E_{max}(SPT^*) \le E_{max}(\pi) \end{array}\right\} \quad \ldots\ldots (2)$$

by influence of EDD rule, and MST rule. Hence all SPT^* are efficient .■

Proposition (2) : If SPT rule , EDD rule and MST rule are identical , then there is only one "non-dominated solution" for the problem (P).

Proof : It is clear . ■

3.1 Some special cases of the problem (P)

The special cases based on lexigraphical order of the cost functions .

Consider the following problems , which are special cases of the problem (P).

1. $1\,||\,Lex(\sum_j C_j, \sum_j T_j\,, T_{max}, E_{max})$ (P1)
2. $1\,||\,Lex(\sum_j C_j, T_{max}, \sum_j T_j\,, E_{max})$ (P2)

Because the cost function $\sum_j C_j$ is more important than other cost functions, the problems P1 and P2 can be solved by the simple algorithem (Lex1):

Algorithem (Lex1)

Step(1) : Order the jobs in non decreasing order of processing times SPT and for the resulting schedule σ calculate the costs functions points $(\sum_j C_j, \sum_j T_j\,, T_{max}, E_{max})$ for P1 and $(\sum_j C_j, T_{max}, \sum_j T_j\,, E_{max})$ for P2

Step(2) : If there exist jobs with equal processing times then order these jobs with EDD rule.

3. $1\,||\,Lex(\sum_j C_j, E_{max}, \sum_j T_j, T_{max})$ (P3)

Also in this problem P3 the cost function $\sum_j C_j$ is more important , so we use the simple hueristic algorithm (Lex2):

Algorithm (Lex2):

Step(1) : Order the jobs in non decreasing order of processing times SPT for the resulting schedule σ calculate the costs functions point $(\sum_j C_j, E_{max}, \sum_j T_j, T_{max})$.

Step(2) : If there exist jobs with equal processing times then order these jobs with MST rule or EDD rule .

4. $1\,||\,Lex(T_{max}, \sum_j C_j, \sum_j T_j, E_{max})$ (P4)
5. $1\,||\,Lex(T_{max}, \sum_j T_j, \sum_j C_j, E_{max})$ (P5)

Clear that the cost function T_{max} is more important than other cost functions so we use the following algorithm (Lex3) :

Algorithm (Lex3)

Step 1 : Solve the problem $1\,||\,T_{max}$ and find $T^* = T_{max}(EDD)$.

Step 2: Calculate $D_i = T^* + d_i\,, i = 1,2,\ldots,n$, let $R = \sum_{j=1}^{n} p_j$, $U = n$, $N = \{1,2,\ldots,n\}$ the set of unscheduling jobs , $\sigma = \sigma(\emptyset)$ the sequence of scheduling jobs.

Step 3: Solve the problem using Smith backward algorithm i.e, find the job $j^* \in N$ such that $D_{j^*} \ge R$ and $p_{j^*} \ge p_j$ for every $j \in N, D_j \ge R$, breake a tie by choosing the job j^* with largest due dates assign job j^* in the position U of σ

Step 4: Set $R = R - p_{j^*}$, $U = U - 1$, $N = N - \{j^*\}$, if $U > 1$ go to step 3.

Step 5: For the resulting sequence calculate the cost functions points
$(T_{max}, \sum_j C_j, \sum_j T_j, E_{max})$ for P4 and $(T_{max}, \sum_j T_j, \sum_j C_j, E_{max})$ for P5.

6. $1 \mid\mid Lex(E_{max}, \sum_j C_j, T_{max}, \sum_j T_j)$ (P6).

7. $1 \mid\mid Lex(E_{max}, \sum_j C_j, \sum_j T_j, T_{max})$ (P7).

In this problem the cost function E_{max} is more important than other cost functions. We give the following algorithm (Lex4) to solve this problem :

Algorithm (Lex4)

Step 1: Sort the jobs according to MST rule and calculate $E^* = E_{max}(MST)$.

Step 2: Set $N = \{1,2,\dots,n\}$ be the set of unscheduled jobs, $\sigma = \sigma(\emptyset)$ be the set of scheduled jobs , $U = 1$,calculate r_j ,where $r_j = \max(d_j - p_j - E^* , 0)$ for every $j \in N$.

Step 3: Find a job $j^* \in \{1,2,\dots,n\}$ such that r_{j^*} is minimum and $r_{j^*} \le C_{U-1}$,where C_{U-1} is the completion time for the job in position $U-1$ and $C_0 = 0$.
If there exist a jobs with equal starting times r_j choose a job j^* with smallest value of p_j , if a tie is still (jobs with equal processing times), choose job j^* with smallest $s_j = d_j - p_j$.

Step 4: Assigne job j^* in the position U of σ, set $N = N - \{j^*\}$, if $N \neq \emptyset$ set $U = U + 1$, go to step 3.

Step 5 : For the resulting sequence, calculate cost functions points
$(E_{max}, \sum_j C_j , T_{max}, \sum_j T_j)$ for P6 and $(E_{max}, \sum_j C_j , \sum_j T_j, T_{max})$ for P7 .

3.2Algorithm(CTTE)for Determination of Approximation set of non-dominated solutions.

For the problem (P), we propose the algorithm (CTTE) to determine the set of approximate solutions (SA). This algorithm decompose the problem (P) in to two subproblems, with objective $(\sum_j C_j, T_{max})$ and $(\sum_j C_j, E_{max})$ and then find the set (SA)

Algorithm CTTE :

Step(0): Set $t = \Delta = \sum p_i$ and $\sigma = (\emptyset)$,calculate $E_{max}(SPT)$ and $E_{max}(MST)$.

Step(1):Set $N = \{1,2,\dots,n\}$, $K = n$, $SA = \emptyset$.

Step(2):Calculate $T_i \quad \forall\, i$ (by Lawler algorithm).

Step(3):Find a job $j \in N$ such that $T_j \le \Delta$, $p_j \ge p_i \ \forall i , j \in N$ and $T_i \le \Delta$
Assign job j in position K of σ , if no job j with $T_j \le \Delta$, 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 $K > 1$ Go to step (2).

Step(5):For the resulting sequence $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(n))$ calculate $(\sum_j C_j(\sigma) , \sum_j T_j(\sigma) , T_{max}(\sigma), E_{max}(\sigma))$, put σ in SA if $\sigma \notin SA$,and non dominated by any solution in SA .

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

Step(7): Put $\Delta = E_{max}(\sigma) - 1$,$N = \{1,2,\dots,n\}, K = 1$, $t = \sum p_j$ and $\sigma = (\emptyset)$
If $\Delta \le E_{max}(MST)$ Go to step (11).

Step(8): Calculate $r_i = \max\{s_i - \Delta , 0\} \ \forall\, i \in N , s_i = d_i - p_i \quad , i = 1,2,\dots,n$

Step(9): Find a job $j \in N$ with minimum (r_j) , $r_j \le C_{K-1}$ and $p_j \le p_i \quad , \forall\, j , i \in N$ break tie with smallest s_j and assign j in position K of σ

Step(10): Set $N = N - \{j\} , K = K + 1$ if $K \le n$ Go to step (9) , otherwise for the resulting sequence $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(n))$ calculate $(\sum_j C_j(\sigma) , \sum_j T_j(\sigma) , T_{max}(\sigma), E_{max}(\sigma))$, put σ in SA if $\sigma \notin SA$, and non dominated by any solution in SA .Go to step (7) .

Step(11): Stop with a set of approximate "non-dominated solutions" (SA) .

3.3 Improvement of CTTE Algorithm (IMCTTE) using Intensification Procedure

A simple technique called (Intensification Procedure) [3] is used to modify and update the set of approximate non dominated solutions (AND) obtained by CTTE algorithm. This procedure constructed of two phases : destruction and construction. In the destruction phase, for each solution s belong to AND, we select randomly d (we called Intesification Perameter) jobs removed from s and sorted in s_r also $(n-d)$ sized partial solution s_p is obtained. The construction phase has d steps. In the first step we have $(n-d+1)$ partial sequences. In this step, the first job in s_r $(i.e\ s_r(1))$ is inserted in each position of the partial sequences and from these partial sequences the non dominated solutions are chosen. In the second step the second job in s_r is inserted in each position of the non dominated solutions obtained by first step, then we have $n-d+2$ partial sequences for each of these solution obtained by first step, and we select again the non dominated solutions from the obtained sequences, and so on. In step d the complete non dominated solutions (AND2) is determined .

Example (1) Consider the following data to illustrate the algorithm (CTTE) and it's modification (IMCTTE) for the problem (P)**:**

$p_i = (2,4,3,1)$, $d_i = (9,8,4,6)$, and $s_i = (7,4,1,5)$,$i = 1,2,3,4$.

Then $E_{max}(SPT) = 17$ and $E_{max}(MST) = 9$, set $t = \Delta = \sum p_j = 10$.

After run the algorithm CTTE and it's modification we have the following results :

Table (1)

No	AND2	CTTE Alg		CEM	
	σ	$F(\sigma)$	Δ	σ	$F(\sigma)$
1	(1,4,3,2)	(39,3,3,17)	$T_{max}(\sigma) - 1 = 2$	(1,4,3,2)	(39,3,3,17)
2	(1,3,2,4)	(43,2,2,17)	$T_{max}(\sigma) - 1 = 1$	(1,3,2,4)	(43,2,2,17)
First part of CTTE algorithm is stop and the second part is run with $\Delta = E_{max}(SPT) - 1 = 16$					
3	(4,1,3,2)	(43,3,3,13)	$E_{max}(\sigma) - 1 = 12$	(4,1,3,2)	(43,3,3,13)
4	(3,1,4,2)	(45,3,3,11)	$E_{max}(\sigma) - 1 = 10$	(3,1,4,2)	(45,3,3,11)
5	(2,1,4,3)	(51,3,3,9)	$E_{max}(\sigma) - 1 = 8$	(2,1,4,3)	(51,3,3,9)
Second part of CTTE algorithm is stop and the IMCTTE algorithm is run with $d = 2$					
6	(3,1,2,4)	(48,2,2,11)		(3,1,2,4)	(48,2,2,11)
7	(2,1,3,4)	(52,2,2,9)		(2,1,3,4)	(52,2,2,9)

Results in table (1) shows that the number of exact "non-dominated solutions" which are found by CEM method is seven solutions, five of them found by CTTE algorithm and the other two solutions are found using the modification part (Intensification Procedure).

4.Multiobjective Local Search Algorithms to find non-dominated solutions For Problem (P)

4.1. Multi objective Genetic Algorithm

The concept of genetic algorithms (GA) was developed by Holland [11]. In genetic algorithm, a population of solutions or chromosomes gives a sequence of transfo- rmations by genetic operators to generate a new population. Two operators are used which are crossover and mutation, crossover operator generate new solution by combining parts of two solutions and mutation generats a new solutions, by a small change in a single solution [14]. Since GA work with a population of solutions, a simple GA can be extended multiobjective problems. With an assurance for moving toward the true Pareto optimal region. GA can be used to find Pareto optimal solutions in one single simulation run .

4.1.1 Non-dominated Sorting Genetic Algorithm 2 (NSGA2) [5]

In the structure of NSGA2, the initial population is generated. Once the population is generated, the population is sorted using the non-domination sorting procedure. This procedure separate the population into fronts. The solutions in the first front are completely non-dominant set in the current population and the second front being dominated by the solutions in the first front only and the front goes so on. Each solution in the each front has a rank. Solutions in first front are given a rank equale to 1 and solutions in second front has rank equal to 2 and so on. In addition to rank value, a new operator called "crowding distance" is calculated for each individual. This operator measure of how close a solution to its neighbors solutions in the same front. A better diversity occurs if a large average of "crowding distance" will result in the population. The binary tournament selection used to select a parents from the population which is based on the rank and crowding distance. A solution is selected compared with other solution if it has lesser rank or larger crowding distance than the other solution. The offsprings are generated to form a new population using crossover and mutation operators. A non domination sorting procedure used again to sort the population with the current population and current offsprings, this sorting is based on and only the best M individuals are selected, where M is the population size. The selection is based on rank and on the crowding distance on the last front.

NSGA2 can be decribed in Figure (1).

4.2 Multiobjective Variable Nieghbourhood Search (MVNS) Algorithm.

Multiobjective Variable Neighborhood Search (MVNS) is first proposed by Geiger [8]. In Geiger's algorithm, randomly chosen a solution from the approximate set of non dominated solutions where no neighborhood search performed, then one arbitrary neighborhood selected and applied to the chosen solution. After applied the neighborhood serach, the approximate set of non dominated solutions is updated. Arroy et al.[3] developed the idea of Geiger using "Intensification Procedure" described in section 3.3 to find the set of "non-dominated solutions" for prblem incloding three performance criteria. We also applied the idea of "Intensification Procedure" and Variable Neighborhood Search and propose the following MOVNS algorithm to find the approximate set of "non-dominated solutions" for problem (P).

MOVNS Algorithm

1. From the initial set of non dominated solutions select four solutions . Let D is the set of these four solutions .
2. Set Total Time = 0 and $EF = \emptyset$, D1=D.
3. Repeat
4. Chose randomly a solution s from D1 and delete it from D1.
5. Generate three neighborhood and applied them to solution s ,then choose randomly one neighborhood solution $\acute{s}$.
6. Use the solution $\acute{s}$ with the Intesification Procedure to find the set AND of non dominated solutions.
7. $EF = EF \cup AND$
8. If Total Time > 3n , go to step (10) .
9. If D1= $\emptyset$, reset D1 (i.e, put D1=D) , go to step (4).
10. Stop
11. Find the non dominated solutions in EF.

5. Computational Experiments:

5.1 Test Problems :

All test problems are conducted on a personal computer intel (R) Core TM i 7 CPU @ 2.50 GHz. ,and 8.00 GB of RAM. To present the efficiency and compared the four algorithms, CTTE algorithm, the Modification CTTE (IMCTTE), proposed MOVNSS algorithm and NSGA2 algorithm, instances with different sizes are considered, the size of these instances are from n=4 to n=100. The processing times for each problem is generated randomly from uniform distribution on the interval [1,100] ,the due dates of each job is drown from uniform distribution on $[(1 - TF - RDD/2) \times t, (1 - TF + RDD/2) \times t]$, where t is total process ing times for all jobs, RDD is the "relative range of due dates", it finds the length of the interval where the due dates are taken. TF the "tardiness factor" find's the relative positions of the centre of the interval between 0 and t , these values of TF and RDD are chosen from the set {0.2,0.4,0.6,0.8,1.0} [2],we generate 10 instances for the combination of TF and RDD.

5.2 Parameter Setting

In our experiment, we find the true Pareto optimal set using complete enumeration method (CEM), because of computation complexity this done for instance up to 7 jobs, the remain expriments use the concept of Referunce set as the approximation of true Pareto optimal set.The reference set is defined here is the non dominated solutions of the union of the "non-dominated solutions" obtained by the four algorithms considered. After conducting several experiments we set parameters of NSGA2 algorithm as follows: The number of generations and initial population size are 100 for $4 \leq n \leq 7$, 150 for $10 \leq n \leq 30$, and 200 for n=40,50,75,100.We use two point order (PMX) crossover with probability 0.7 and the mutation is swap mutation with probability 0.3.We use three methods to find three initial solutions in the initial population, the three methods are SPT , EDD and MST rules, the other solutions are generated randomly. For MOVNS we set d (Intensification Parameter) is equal to 3, and the algorithm stoped after 3n seconds.

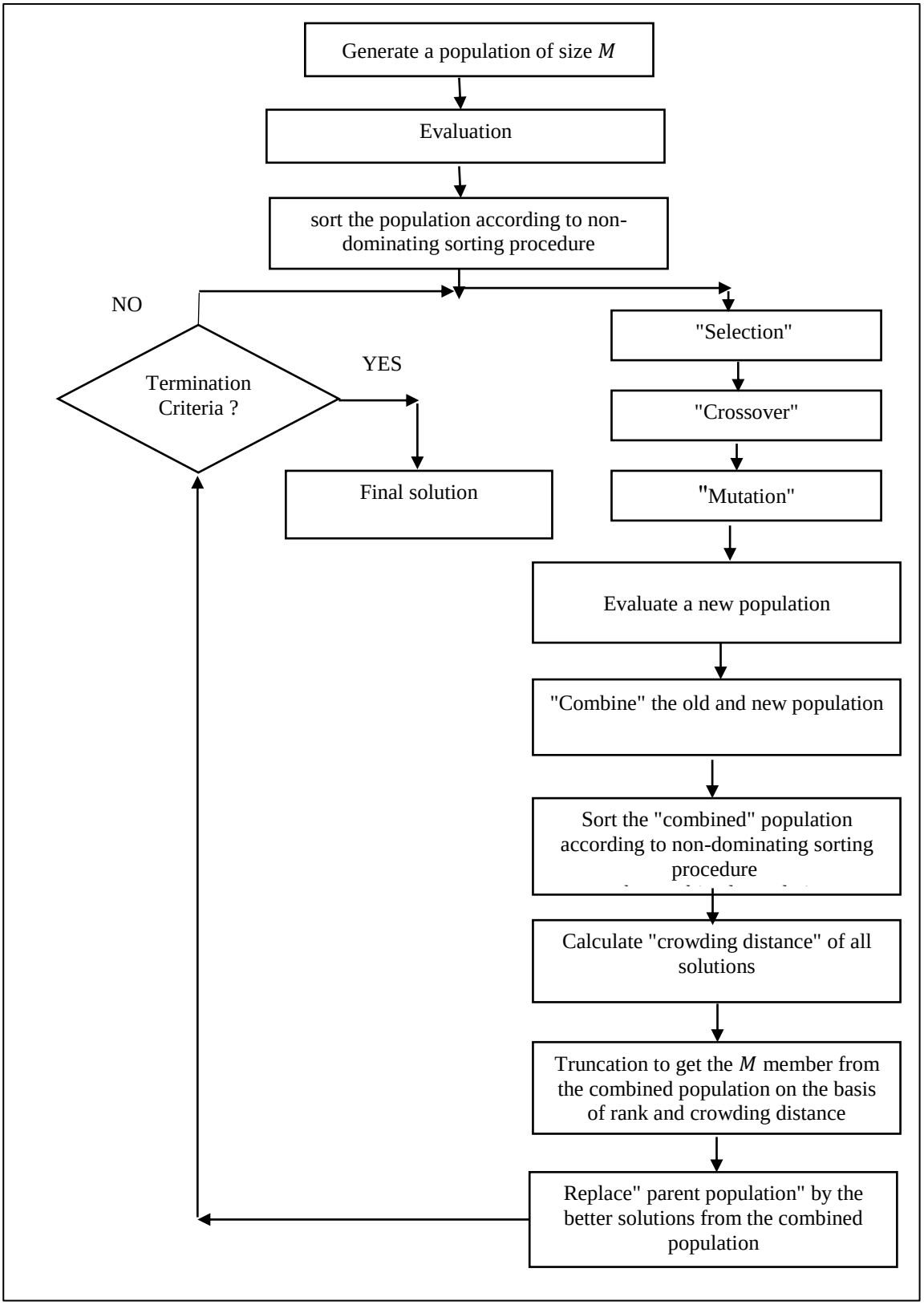
5.3 Performance Comparison:

For the comparison of the three algorithm we use the Cardinal Measure $C(REF, A)$ [3],where $C(REF, A) = |REF \cap A|$,which compute the number of non dominated solutions obtained by algorithm A where A is the algorithm that needed to measure its performance, that belong to the referunce set REF, $|\ \ |$ denote the number of non dominated solutions.

5.4 Comparison Results

To compare the four algorithms, proposed algorithm (CTTE), and it's Improvement IMCTTE, NSGA2 algorithm and MOVNS algorithm. We use cardinal measure as performance comparison measure between reference sets and the set of "non-dominated solutions" obtained by each algorithm. There are two types of reference sets considered in this study, the first considered as true set of "non-dominated solutions" for problem (P) calculated by CEM algorithm which can not be obtained for problem instances of size greater than 7 jobs due it's computation times, the second reference set considered a

Figure(1) Flowchart of NSGA2


Generate a population of size M
Evaluation
sort the population according to non-dominating sorting procedure
NO
Termination Criteria ?
YES
Final solution
"Selection"
"Crossover"
"Mutation"
Evaluate a new population
"Combine" the old and new population
Sort the "combined" population according to non-dominating sorting procedure
Calculate "crowding distance" of all solutions
Truncation to get the M member from the combined population on the basis of rank and crowding distance
Replace" parent population" by the better solutions from the combined population

approximate set of "non-dominated solutions" for problem (P) obtained by calculate the non dominated solutions of the union of the sets of "non-dominated solutions" obtained by the four algoriothms (CTTE, IMCTTE, MOVNS, NSGA2). Table (2) contains the average number of "non-dominated solutions" in reference sets M(RF) and each of the CTTE, IMCTTE, MOVNS, NSGA2 algorithm M(CTTE), M(IM), M(VN), M(NS). Also the average CPU Time M(T) for the CTTE, IMCTTE, MOVNS, NSGA2 algor- ithms. Table (3) contains the average values of cardinal measure. We notes the following results :

1. Table (2) shows that the number of "non-dominated solutions" obtained by IMCTTE algorithm are more than those obtained by CTTE algorithm.
2. Table (3) shows that the number of "non-dominated solutions" obtained by IMCTTE algorithm that belong to reference set is more than those obtained by CTTE algorithm that belong to reference set.
3. Table (2) shows that the MOVNS algorithm give more "non-dominated solution"s than other algorithms , and the two algorithms MOVNS and NSGA2 are close together for small size problems.
4. Table (3) shows that MOVNS and NSGA2 algorithms give better contribution of the "non-dominated solutions" in reference set than other algorithms , and MOVNS has better performance than NSGA2 for problem sizes $n \leq 20$, and $n = 100$. Where NSGA2 algorithm is better than MOVNS algorithm for problem sizes $30 \leq n \leq 75$

Table (2) The average of : CPU Time M(T) and number of "non-dominated solutions" M(RF) obtained by CEM(for $4 \leq n \leq 7$, and REF for $n \geq 10$),CTTE, IMCTTE, MOVNS, and NSGA2 algorithms.

n	M(RF)	M(CTTE)	M(T)	M(IM)	M(T)	M(VN)	M(T)	M(NS)	M(T)
4	4.3	3.6	0.006	4.2	0.006	4.3	12	4.3	57.2
5	8	4.8	0.006	5.9	0.012	8	15	8	57.1
6	11.1	5.5	0.008	7.6	0.016	11.1	18	10.9	56.5
7	22.9	6.8	0.010	9.2	0.033	22.9	21	20	56.2
10	37.9	8.5	0.018	13.6	0.081	37.8	30	29.8	182.9
20	147	17.3	0.071	27.8	1.086	135.3	60	93	179.0
30	198.5	23.2	0.15	31.1	4.469	175.9	90	114.8	178.0
40	308.1	36.4	0.37	46.9	20.75	245.1	120	181.5	635.2
50	292.7	42.7	0.69	62.8	68.26	225.8	150	179.3	638.5
75	404.8	69.1	1.9	87.9	249.7	271	258.6	190	639.3
100	519.5	98.1	4.5	129.7	329.1	353.8	356.1	191.7	627.4

Table (3) The average values of Cardinal Measure obtained by: Reference set M(RF) (CEM for $4 \leq n \leq 7$ and Reference set for $n \geq 10$) and by each of the CTTE, IMCTTE, MOVNS, and NSGA2 algorithms.

n	M(RF)	M|CTTE∩RF|	M|IM∩RF|	M|VN∩RF|	M|NS∩RF|
4	4.3	3.5	4.1	4.3	4.3
5	8	4.5	5.5	8	8
6	11.1	5.2	6.1	11.1	10.9
7	22.9	5.7	6.3	22.9	19.1
10	37.9	7.6	9.6	37.7	24.6
20	147	15	15.4	85	65
30	198.5	21.1	21.4	85.8	98.3
40	308.1	34.3	36	110.4	165.4
50	292.7	41	43.2	83.8	169.2
75	404.8	67.7	71	148.6	187
100	519.5	97.3	107.4	221.1	191.6

M: The mean value, RF: Reference set, T: CPU Time in seconds, VN: MOVNS, NS: NSGA2, IM: IMCTTE

6. Conclution

The main problem in this study is to find the set of "non-dominated solutions" for problem of minimization four cost functions, total completion times, total tardiness, maximum tardiness, and maximum earliness simultanuasly and some special cases including study in lexigraphical order minimization of our problem . For the simultanuase minimization of our problem we propose the CTTE algorithm that find the set of "non-dominated solutions" and try to improve the non-dominated solutions set obtained by CTTE algorithm using proposed IMCTTE. Also we propose MOVNS algorithm based in the variable neighborhood and Intensification Procedure ideas. We compare proposed algorithms with NSGA2 algorithm. According to the results we find that our proposed CTTE algorithm give resonable results specially in small size problems and the IMCTTE algorithm give better performance than original CTTE algorithm. Also we find that proposed MOVNS algorithm give better performance than CTTE and IMCTTE algorithms for all problem instancs, and better than NSGA2 specially for small size problems .

7. References

[1]. Abdul-Razaq, T.S ,Mahrooz.Z. A, (2015) "Minimizing the total complition times ,the total tardiness and the maximum tardiness"ibn AL-Haitham J for Pure & Applide Sci . Vol.28 (2) .

[2]. Abdul-Razaq, T.S,Ibraheem.S.J (2014) "Single machine with maximum Earliness And maximum Tardiness" International Jornal of Mathematical Archive-5(9),2014,79-82.

[3]. Arroyo J.E, Ottoni R.S,Oliveira A .P,(2011)"Multi-objective Variable Neighborhood Search Algorithm for a single Machine Scheduling Problem with Distinct due Windows" Electronic Notes in Theoretical Computer Science 281(2011)5-19 .

[4]. Baker, K. R., & Smith, C. J. (2003). A multiple criterion model for machine scheduling. Journal of Scheduling, 6, 7–16.

[5]. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., A (2002)," fast and elitist multi-objective genetic algorithm: NSGA2-II, IEEE Transactions on Evolutionary Computation 6(2) 182-197.

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

[7]. Fonseca.C.M,Fleming,P.J(1995)."An Overview of Evolutionary Algorithms in Multi-Objective Optimization".Evolutionary Computation,3(1):1-16.

[8]. Geiger,M.J.,(2004),"Randomized variable neighborhood search for multiob- jective optimization", 4 th EU/ME: Design and Evaluation of Advanced Hybrid Meta-Heuristics,34-42.

[9]. Hoogeveen, J.A., (2005), "Invited review Multicriteria Scheduling" , European Journal of operational Research 167,592-623.

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

[11]. J.H. Holland (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press (1992).

[12]. Murata.T,Ishibuchi.H.(1995) " MOGA : Multi-Objective Genetic Algorithms". International Conference on Evolutionary Computation , pp.289 -294. Nov- ember 1995.

[13]. Najar. A (1996)." A combined branch-and-bound and genetic algorithmbased approach for a flowshop scheduling problem ". Annals of Operations Research 63(1996)397-414.

[14]. Rahimian.M, Soltani.R,Tahmasbi.A,Mahmoodi.A,Alizadeh.S.Z (2012) "Multi-criteria sequence dependent single machine scheduling using genetic and simulated annealing algorithms",International Journal of Emerging Technology and Advanced Engineering,(ISSN 2250-2459, Volume 2, Issue 8, August 2012.

[15]. Schaffer, J.D (1985)."Multiple Objective Optimization with vector Evaluated Genetic Algorithm".In Grefenstette,JJ.,editor,Proceeding of an International conference on Genetic

Algorithm and their applications,pages 93-100,sponsored by Texas Instrument and U.S.Navy Center for applied Research in Artificial Intelligence.

[16]. Sen,T & S.K Gubta, (1983)." A Branch-and-Bound Procedure to Solve a Bicriterion Scheduling Problem" IIE Trans.,15,1,84-88.

[17]. Sioud A, Gravel M, Gagné C. (2010). Constraint based scheduling in a genetic algorithm for the single machine scheduling problem with sequence dependent setup times. In: ICEC'2010: proceedings of the international conference on evolutionary computation, 137-45.

[18]. Smith, W.E., (1956), " Version optimization for single stage production", Naval Res.Logistics Quarter, 3, 59-66.

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

[20]. Vanwassenhove, L.N., and Gelders, L.F., (1980), "Solving a bicriterion scheduling problem", European Journal of Operational Res. 4, 42-48.

[21]. Zitzler.E, (1999).” Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, ETH,Zurich, Switzerland, 1999. Shaker Verlag, Germany.