Single Machine Scheduling To Minimize a Function of Square Completion Time and Maximum Tardiness Simultaneously

[1]Tariq Salih Abdul-Razaq And [2]Haidar younis kawi
The University of Al-[1]Mustansiriya, College of Science,Department of Mathematics
[2]The University of Al-Qadisiya ,College of Computer Science and Mathematics,Department of Mathematics

الخلاصة

في هذه الدراسة ولتصغير دالة الكلفة لمعيارين والحاصلة من جدولة n من الاعمال على ماكنة واحدة درست المسألة:

تصغير الدالة $F(\sum C_i^2, T_{max})$ حيث ان T_{max} هي regular measure في هذه المسألة اقترحنا بعض الخوارزميات لايجاد الحل الامثل في حالة الـ (hierarchical) والحلول الكفوءة في حالة الـ (simultaneous) . وكذلك اقترحنا خوارزمية للـ (BAB) لايجاد الحل الامثل للمسألة (P4) . وقدمنا ايضاً حوارزمية B لايجاد الحل الامثل للمسألة (P4) ولكن بطريقة اسرع من خوارزمية(BAB) . وقدمنا حسابات الاختبارات لخوارزميات BAB و B والتي تم تنفيذها على مجموعة كبيرة من المسائل.

ABSTRACT

In this study, to minimize a function of two cost criteria for scheduling n jobs on a single machine , the problem is discussed :

" Minimizing a function of total square completion time and maximum tardiness simultaneously".

For this problem we proposed some algorithms to find exact(optimal) solution for hierarchical case and efficient (pareto optimal) solutions for simultaneous case, Also we proposed branch and bound algorithm to find exact solution for sum of total square completion time and maximum tardiness, and present algorithm B to find exact solution in a fast way with respect to (BAB) method. We present computational experience for the (BAB) method and algorithm(B) on a large set of test problems.

INTRODUCTION

It is well known that the optimal solution of single objective models can be quite different if the objective is different (for instance, for the simplest model of one machine, without any additional constraint, the rule SPT is optimal to minimize flow time but the rule EDD is optimal to minimize the maximal tardiness T_{max}).

In fact, often each particular decision maker wants to minimize a given criterion.

Recently, research on more than one criterion scheduling has increased. Since real life scheduling problems may require the decision maker to consider a number of criteria before arriving at any decision. Nagar et al. (20) in their detailed literature survey of multiple and bi-criteria problems in scheduling point out the importance of this subject.

Because, the one-machine problem provides a useful laboratory for the development of ideas for heuristics and interactive procedure that may prove to be useful in more general models. We consider the one-machine case in this study.

Multi-Criteria Scheduling:

In general, multiple-criteria scheduling refers to the scheduling problem in which the advantages of a particular schedule are evaluated using more than one performance criterion. The managerial relevance of considering multiple criteria for scheduling has been cited in the production and operations management literature since the 1950's. Smith (1956)(22) shows that the choice of a criterion will affect the characteristics of a "best schedule"; different optimizing criteria will result in very different schedules. Van Wassenhove and Gelders (1980)[25] and provide evidence that a schedule that performs well using a certain criterion might yield a poor result using other criteria. Hence, lack of consideration of various criteria may lead to solutions that are very difficult to implement in practice. Although the importance of multi-criteria scheduling has been recognized for many years (French, 1982(7); Nelson et al., 1986(21); George S., and Paul S. 2007(8), little attention has been given in the literature to this topic. From the problem complexity perspective, the multiple-criteria problem becomes much more complex than related single-criteria counterparts (Lenstra et al., 1975(18) Nagar et al. (1995)(20) reviews the problem in its general form whereas Lee and Vairaktarakis (1993)(16) review a special version of the problem, where one criterion is set to its best possible value and the other criterion is tried to be optimized under this restriction. Hoogeveen (2005)(11) studies a number of bi-criteria scheduling problems.Also, there are some papers about this object (Cheng et al. 2008(5), and Azizoglu et al. 2003 (1).

Approaches for Multi-Criteria Problems:

In literature there are two approaches for the bi-criteria problems: the hierarchical approach and the simultaneous approach. In the hierarchical approach, one of the two criteria is considered as the primary criterion and the other one is considered as the secondary criterion. The problem is to minimize the primary criterion while breaking ties in favor of the schedule that has the minimum secondary criterion value. The studies by Chang P. and Su L.(2001)(3) and Chen W., et al.(1997)(4) are examples of hierarchical minimization problems with earliness and tardiness costs. The computational complexity results in hierarchical minimization are reviewed in Lee and Vairaktarakis (1993)(17). In the simultaneous approach there are two types ,the first one typically generates all efficient schedules and selects the one that yields the best composite objective function value of the two criteria .The second is to find sum of these objectives .Several scheduling problems considering the simultaneous minimization of various forms of earliness and tardiness costs have been studied in the literature (see, e.g. Hoogeveen, (1995)(12); Moslehi, et al. (2005)(19)) .

Basic definitions:

Definition(1):(14) *The term "optimize" in a multi-objective decision making problem refers to a solution around which there is no way of improving any objective without worsening at least one other objective.*

Definition(2) (14) *Suppose we have a problem P ,any schedule* $S \in \delta$ *(where* δ *is the set of all schedules) is said to be feasible if it satisfies the constraints of the problem P.*

Definition(3): (1). *A schedule S is said to be efficient if there does not exist another schedule* S' *satisfying* $f_i(S') \leq f_i(S)$ *, i=1,…,k with at least one of the above holding as a strict inequality. Otherwise S is said to be dominated by* S'.

Definition(4): (20) *A measure of performance is said to be regular if it is a non-decreasing function of job completion times and the scheduling objective is to minimize the performance measure. Examples of regular measures are job flowtime (*$\bar{F}$*), schedule makespan (*C_{max}*) and tardiness based performance measures.*

Definition (5): (11) *The function F(f,g) is said to be non-decreasing in both argument ,if for any pair of outcome value (x,y) of the functions f and g ,we have* $F(x,y) \leq F(x+A,y+B)$ *for each pair of non-negative value A and B.*

Theorem (1): (11) If the composite objective function F(f,g) is non-decreasing in both argument ,then there exists a pareto optimal schedule that minimize F.■

Basic Scheduling Concepts

We start with introducing some important notation where we concentrate on the performance criteria with out elaborating on the machine environment etc. We assume that there are n jobs, which we denoted by $j_1,\ldots,j_n$ these jobs are to be scheduled on a set of machines that are continuously available from time zero on words and that can handle only one job at a time .

In this paper, we only state here the notation that is used for single machine , jobs J_i(i=1,…,n) has:

N: set of jobs.

n: The number of jobs in a known sequence.

P$_j$: which means that it has to processed for a period of length p_j .
d$_j$: a due date ,the date when the jobs should ideally be completed , the completion of job after its due date is allowed ,but a penalty is incurred . When the due date absolutely must be met , it is referred to as deadline $\bar{d}_j$, and when due date is constant for all jobs ,then called common due date.

- The completion time C_j
- The lateness $L_j=C_j - d_j$
- The tardiness $T_j=\max\{0, C_j-d_j\}$

For a given schedule σ we compute.

- $C_{max}(\sigma) = \max_j(C_j)$
- $L_{max}(\sigma)=\max_j(L_j)$
- $T_{max}(\sigma)=\max_j(T_j)$

Fundamental Results and Algorithms:

Theorem (2)(Smith_1956)(22). The $1/ / \Sigma C_i$ problem is minimized by sequencing the jobs according to the shortest –processing-time (SPT) rule, that is, in order of non-decreasing p_i.■

Theorem(3)(Jackson 1955)(13). The $1/ / L_{max}$ problem is minimized by sequencing the jobs according to the earliest-due- date (EDD) rule, that is, in order of non-decreasing d_i.■

Theorem(4)(Lawler 1973)(16).The $1//f_{max}$ problem, f_{max} is minimized as follows: while there are unassigned jobs, assign the job that has minimum cost when scheduled in the last unassigned position in that position.■

Hoogeveen and Van de Velde (12) provide a generalization to the case that the two criteria are $\sum C_j$ and f_{max} where f_{max} is regular cost function.

Van Wassenhove and Gelder (24)propose a pseudo-polynomial algorithm for finding all efficient schedules with respect to $\sum C_j$ and L_{max} .Their algorithm searches all possible L_{max} values .Since a given L_{max} value imposes job dead line $\bar{d}_j$,the algorithm of Smith (21) is used to solve the corresponding $1/\bar{d}_j / \sum C_j$ problem.

The Problem Classification:

In this paper, we adopt the terminology of Graham ,Lawler ,and Rinnooy Kan _(1979) [9] to classify scheduling problems.

Suppose that m machines M_i (i=1,…,m) have to process n jobs J_j ($_{j=1,\dots,n}$) . A schedule problem type can be specified using a three-field classification α/β/γ composed of the machine environment, the job characteristics, and the optimality criterion .

Minimizing Total Square Completion Time

This section deals with the Quadratic problem of scheduling jobs on a single machine such that the sum of the square of the weighted completion times of jobs is minimized(i.e. $1 // \sum_{i=1}^{n} w_i C_i^2$ problem) . Relatively little work has been done on problems involving a quadratic measure of performance for scheduling a single machine. The single machine scheduling problem with the objective of minimizing the sum of squares of the job completion times has been studied by Townsend (1978)(24), Bagga and Kalra (1980)(2), Gupta and Sen (1984)(10), and Szwarc, Posner, and Liu (1988)(23). Townsend (24) first formulated the problem and presented a branch-and-bound search method to solve it. Bagga and Kalra (2) improved the method by providing conditions for precedence among set of jobs. If w_i =1 for every i , then the resulting problem $1 // \sum_{i=1}^{n} C_i^2$ is solved by the following proposition.

Proposition(1):(15)The SPT rule gives an optimal value for $1// \sum_{i=1}^{n} C_i^2$ problem.

Minimizing Total Square Completion Time and Maximum Cost

Now, we will consider the bi-criteria single machine problem concerns the simultaneous minimization of the performance measure total square completion time $\sum_{i=1}^{n} C_i^2$ and maximum cost f_{max} (i.e. $1//F(\sum_{i=1}^{n} C_i^2, f_{max})$ problem).Maximum cost is defined as $\max_{1 \le i \le n} \{f_i(C_i)\}$, where each f_i denotes an arbitrary regular or irregular cost function for job i; regular means that $f_i(C_i)$ does not decrease when C_i is increased .

The $1// F (\sum_{i=1}^{n} C_i^2, f_{max})$problem is described as follows. A set of n independent jobs has to be scheduled on a single machine that is continuously available from time zero on wards and that can process at most one job at a time. Each job J_j *(* j *=1, ..., n)* requires an uninterrupted positive processing time p_j and has a due date d_j. Without loss of generality, we assume that the processing times and due dates are integral. A *schedule σ* specifies for each job when it is executed while observing the machine availability constraints. Hence, a schedule σ defines for each job J_j its square of completion time $C_j^2(\sigma)$, which we sometimes simply write as C_j^2 .

The bi-criteria problem that we consider concerns the simultaneous minimization of the performance measures ***total square completion time*** and ***maximum cost*** f_{max}.

Hoogeveen and Van de Valde (12) proved that $1//F(\sum C_i, f_{max})$ problem is solved in polynomial time ,Van Wassenhove and Gelders(25) solved $1//F(\sum C_i, T_{max})$ problem, Emmons (6)addresses the hierarchical problem $1/ /Lex(f_{max}, \sum C_i)$,where f* denotes the optimal solution value of the 1// f_{max} problem, which is solved in $O(n^2)$ time by Lawler algorithm(16) .
Let $f_{max} = T_{max}$ in our study , since criterion T_{max} is a particular case of the function f_{max} .

Now , consider the following two problems:

$1//Lex(\sum_{i=1}^{n} C_i^2, T_{max})$ problem ,and $1//Lex(T_{max}, \sum_{i=1}^{n} C_i^2)$ problem.

The first problem $1//Lex(\sum_{i=1}^{n} C_i^2, T_{max})$

This problem can be written as:

$$\left.\begin{array}{l} \text{Min } T_{max} \\ \text{s.t} \\ \sum_{i=1}^{n} C_i^2 = C^* \quad \text{where } C^* = \sum C_i^2 \text{(SPT)} \end{array}\right\} \ldots \text{(P1)}$$

Algorithm for problem(P1):

Setp(0): Order the jobs by SPT rule and calculate $\sum_{i=1}^{n} C_i^2$ and T_{max}.

Step(1): If there exist a tie(jobs with the same processing times) order these jobs by EDD rule to minimize T_{max} .

Note that the problem (P1) can be written as:

$$1/ \sum_{i=1}^{n} C_i^2 = C^* /T_{max}.$$

Example-1: Consider the problem (P1) with the following data:

i	1	2	3	4	5
P_i	2	2	5	9	5
d_i	10	10	9	19	5

It is clear that the SPT rule is optimal for problem (P1).

i	1	2	3	5	4
P_i	2	2	5	5	9
d_i	10	10	9	5	19
C_i	2	4	9	14	23
C_i^2	4	16	81	196	529
T_i	0	0	0	9	4

Hence the SPT_1 schedule (1,2,3,5,4) with ($\sum_{i=1}^{n} C_i^2$,T_{max})= (826,9).

But the SPT_2(break a tie of job 3 and 5) the schedule (1,2,5,3,4) with ($\sum_{i=1}^{n} C_i^2$,T_{max})= (826,5).

The second problem 1//Lex(T_{max} , $\sum_{i=1}^{n} C_i^2$)

This problem can be written as:

$$\left.\begin{aligned} &\text{Min } \sum_{i=1}^{n} C_i^2 \\ &\quad \text{s.t.} \\ &\quad T_{max}=T^* \text{ where } T^*=T_{max}(EDD) \end{aligned}\right\} \ldots (P2)$$

Also the problem (P2) can be written as: 1/ $T_{max}=T^*$ /$\sum_{i=1}^{n} C_i^2$,

which is equivalent to the problem 1/$\bar{d}_j$/$\sum_{i=1}^{n} C_i^2$ where $\bar{d}_j=d_j+T^*$.

Its clear that problem (P2) can be solved by "Smith backward" algorithm.

Algorithm(A) for problem(P2):

Step(0): Order the jobs by EDD rule and calculate $T_{max}(EDD) = T^*$.

Step(1): Find $\bar{d}_j=d_j+T^*$ $\forall j \in N$,N={1,…,n} unscheduled jobs ,and σ = (φ) for schedule jobs .

Step(2): Let t = $\sum_{j=1}^{n} p_j$

Step(3): Find a job j*∈ N satisfy $\bar{d}_{j^*} \geq t$ (if there exist a tie choose the job j* with largest processing time).

Step(4): Set t= t –p_{j^*} , N =N-{j*} , σ =(σ(j*) ,σ), if N= φ go to step (5), else go to step (3).

Step(5): Calculate $\sum_{i=1}^{n} C_i^2(\sigma)$ and $T_{max}(\sigma)$.

It is clear from the algorithm (A) ,we are interested in the minimization of 1//Lex(T_{max} , $\sum_{i=1}^{n} C_i^2$) problem. Since the SPT schedule minimizes $\sum_{i=1}^{n} C_i$ and $\sum_{i=1}^{n} C_i^2$ for the single machine problem [see proposition(1)] .Hence 1//Lex(T_{max} , $\sum_{i=1}^{n} C_i^2$) problem is equivalent to 1//Lex(T_{max}, $\sum_{i=1}^{n} C_i$) problem . The later problem is a particular case of the 1//F(T_{max} , $\sum_{i=1}^{n} C_i$) problem which is solved by Van Wassenhove and Gelders [25].This means that we can solve 1//Lex(T_{max}, $\sum_{i=1}^{n} C_i^2$) problem and this can be done by algorithm (A).

Example-2: Consider the problem (P2) with the following data:

i	1	2	3	4	5
P_i	5	8	8	10	9
d_i	11	8	14	10	10

EDD rule gives the schedule (2,4,5,1,3) with T_{max}(EDD)=26=T*

i	2	4	5	1	3
P_i	8	10	9	5	8
d_i	8	10	10	11	14
C_i	8	18	27	32	40
T_i	0	8	17	21	26

Since $\bar{d}_j$ =d_i +T* ,hence by using algorithm(A),

$\bar{d}_1$ =37 , $\bar{d}_2$ = 34 , $\bar{d}_3$ = 40 , $\bar{d}_4$ = 36 , $\bar{d}_5$ = 36

j	t	j*
1	40	3
2	32	4
3	22	5
4	13	2
5	5	1

Hence the schedule (1,2,5,4,3) with (T_{max}, $\sum_{i=1}^{n} C_i^2$) =(26, 3302).

In the following section we consider the general problem F(($\sum_{i=1}^{n} C_i^2$,T_{max}).

Total Square Completion Time and Maximum Tardiness

In this section we will try to find an efficient (pareto optimal) solutions for 1/ /F($\sum_{i=1}^{n} C_i^2$,T_{max}) problem.

The 1/ /F($\sum_{i=1}^{n} C_i^2$,T_{max}) problem can be written as:

$$\left.\begin{array}{l} \text{Min } \sum_{i=1}^{n} C_i^2 \\ \text{s.t.} \\ T_{max} \le T \quad \text{where } T \in [\, T_{max}(\text{EDD}),\, T_{max}(\text{SPT})] \end{array}\right\} \ldots (P3)$$

Theorem (5)[11]: Consider the composite objective function F with $F(\pi) = F(f_1(\pi),\ldots,f_k(\pi))$,where F is non- decreasing in all performance criteria f_k . There is a pareto optimal schedule with respect to $f_1,\ldots,f_k$ that minimizes the function F.■

Note that in the following proposition H= $\sum_{i=1}^{n} C_i^2$ and T =$T_{max.}$.

Proposition(2):(15) *There exists an efficient sequence for problem (P3) that satisfy the SPT-rule.*

Note that an analogous proposition for the EDD rule does not hold in general as shown by the following example :

i	1	2	3	4
P_i	2	3	1	2
d_i	3	4	5	6

SPT* sequence (3,1,4,2), H(SPT*) = 99, T(SPT*) = 4
EDD sequence(1,2,3,4), H(EDD) = 129, T(EDD) = 2
SPT* is efficient by Proposition (2.1).
EDD is not efficient since it is dominated by sequence (3,1,2,4) with H = 110 and T = 2.

It is clear that the $1//F(\sum_{i=1}^{n} C_i^2, T_{max})$ problem originates from $1//\sum_{i=1}^{n} C_i^2$ problem and $1//T_{max}$ problem. Both problems are solvable in O(n log n) time.

In order to find the set of pareto optimal points , we solve the problem of minimizing $\sum_{i=1}^{n} C_i^2$ subject to $T_{max} \le T^{**}$, where T^{**} corresponds to the T_{max} value of a pareto optimal point.

The next algorithm solve problem $1//F(\sum_{i=1}^{n} C_i, T_{max})$.

Algorithm(B) for (P3):

Step(0): Compute T_{max}(EDD) ,and T_{max} (SPT) ;let k=1 , T_{max} (SPT)=T**.

Step(1): Solve $1/T_{max} \le T^{**} / \sum_{i=1}^{n} C_i^2$ by algorithm(A) for (P2) ; this produces the k pareto optimal schedule $\sigma^{(k)}$, and the k pareto optimal point $(\sum_{i=1}^{n} C_i^2(\sigma^{(k)}), T_{max}(\sigma^{(k)}))$.

Step(2): T**=T** -1, k=k+1.

Step(3): If T**< T_{max}(EDD) stop, else ,go to step (1).

Example-3: Consider the problem (P3) with the following data:

i	1	2	3	4	5
P_i	3	1	7	7	10
d_i	4	12	14	8	10

T_{max}(EDD)=14 ,T_{max}(SPT)=18=T**.
Now, by proposition (2.1) ,SPT rule gives efficient schedule(2,1,3,4,5) then the first point is (1246,18).
T**=18-1=17
Now we will solve 1/ $T_{max} \leq 17$ $/\sum_{i=1}^{n} C_i^2$ by algorithm(A) for (P2) .Hence

$\bar{d}_1$ =21 , $\bar{d}_2 = 29$, $\bar{d}_3 = 31$, $\bar{d}_4 = 25$, $\bar{d}_5$ =27 .

j	t	j*
1	28	3
2	21	5
3	11	4
4	4	1
5	1	2

Hence the second efficient schedule (2,1,4,5,3), and the second point is (1363,14).
T**=14-1=13< T_{max}(EDD). Stop.
It is clear that from the example that the EDD schedule (1,4,5,2,3) with (1734,14) is not efficient.

The 1//$\sum_{i=1}^{n} C_i^2$ +T_{max} problem:

In this section we decompose the 1//$\sum_{i=1}^{n} C_i^2$ +T_{max} problem into two subproblems with a simpler structure , and state some results which help us in solving it.

This problem can be written as:

$$\left.\begin{aligned}
&M_1 = \min_{\sigma \in s} \{ \sum_{i=1}^{n} C_i^2 + T_{max}(\sigma) \} \\
&\text{s.t.} \\
&c_{\sigma(i)} \geq p_{\sigma(i)} \\
&c_{\sigma(i)} = C_{\sigma(i-1)} + P_{\sigma(i)} \\
&T_{\sigma(i)} \geq C_{\sigma(i)} - d_{\sigma(i)} \\
&T_{\sigma(i)} \geq 0
\end{aligned}\right\} \quad \ldots (P4)$$

This problem can be decomposed into two subproblems (SP1) and (SP2) .

$$\left.\begin{array}{l} V_1=\min\limits_{\sigma\in s}\ \sum\limits_{i=1}^{n} C_{\sigma(i)}^2 \\ \text{s.t.} \\ c_{\sigma(i)} \geq p_{\sigma(i)} \quad i=1,\ldots,n \\ c_{\sigma(i)} = C_{\sigma(i-1)}+P_{\sigma(i)} \quad i=2,\ldots,n \end{array}\right\} \ldots(SP1)$$

$$\left.\begin{array}{l} V2=\min\limits_{\sigma\in s}\ \{\max\{T_{\sigma(i)}\}\} \\ \quad \text{s.t.} \\ \quad T_{\sigma(i)} \geq C_{\sigma(i)} - d_{\sigma(i)} \quad i=1,\ldots,n \\ \quad T_{\sigma(i)} \geq 0 \qquad\qquad i=1,\ldots,n \end{array}\right\} \ldots(SP2)$$

Theorem(6)(20) :

V1+V2≤ M1 where V1 ,V2 ,and M1 are the minimum objective function values of (SP1),(SP2), and (P4) respectively .■

Some Special Cases for the Problem (P4).

Case(1): If for every schedule $C_i \leq d_i\ \forall\ i\in N$ then SPT rule gives an optimal value for (P4).

Proof: Since $C_i \leq d_i$ then $T_i=0\ \forall\ i\in N$ $T_{max}=0$.

Hence the problem (P4) reduce to $1//\sum\limits_{i=1}^{n} C_i^2$ problem .Then by proposition(1) SPT rule gives optimal value .■

Case(2): If $p_i=p\ \forall\ i\in N$ then EDD rule gives an optimal value for (P4).

Proof: If $p_i=p\ \forall\ i\in N$ then $\sum\limits_{i=1}^{n} C_i^2$ is constant for every sequence ,since EDD rule gives minimum value for T_{max} , then EDD rule gives optimal value for (P4).■

Case(3): If $d_i=d\ \forall\ i\in N$ then SPT rule gives an optimal value for (P4).

Proof: If $d_i=d\ \forall\ i\in N$ then T_{max} is constant for every sequence ,since SPT rule gives minimum value for $\sum\limits_{i=1}^{n} C_i^2$, then SPT rule gives optimal value for (P4).■

Case(4): If the due date is agreeable (i.e.$p_1\leq\ldots\leq p_n$ and $d_1\leq\ldots\leq d_n$) then the SPT and EDD rule give an optimal schedule .

Proof: Since $p_1\leq\ldots\leq p_n$ then $\sum\limits_{i=1}^{n} C_i^2$ is minimum value , and at the same time $d_1\leq\ldots\leq d_n$ then T_{max} is minimum value ,then $\sum\limits_{i=1}^{n} C_i^2 + T_{max}$ is the minimum value .■

Heuristic to calculate upper bound (UB) for the problem (P4).

To calculate upper bound (UB_T)order the jobs by SPT rule and then calculate $\sum_{i=1}^{n} C_i^2$ and $T_{max.}$

Derivation of lower bound (LB)

To calculate a lower bound (LB) apply theorem (6).

Example-4 : Consider the problem (P4) with the following data:

i	1	2	3	4
P_i	3	4	8	7
d_i	12	4	10	7

The SPT rule gives the schedule (1,2,4,3) where $\sum_{i=1}^{n} C_i^2$ =738,

The EDD rule gives the schedule (2,4,3,1) where T_{max}=10

Hence the LB=738+10=748.

Note that the exact solution for problem (P4) obtained by (BAB) method. The optimal schedule is (1,2,4,3) with $\sum_{i=1}^{n} C_i^2$ +T_{max}=750.

The lower bound of each node in the solution search tree are written against the nodes of the tree. To find the optimal solution for (P4), we applied the methods for lower and upper bounds that will be used in BAB algorithm. Where (BAB) Branch and bound method can be used for solving many combinatorial optimization problems. These procedures can be conveniently represented as a search (scheduling, branching) tree whose nodes correspond to subsets of a feasible solution. To minimize an objective function of a particular scheduling problem, first an upper bound UB of the minimum of this objective function is needed. A branching rule is used to partition feasible solutions at a node into subsets and a bounding rule calculates a lower bound LB on the value of each solution in a subset.

Computational experience

An intensive work of numerical experimentations has been performed. We first present how instances (tests problem) can be randomly generated .

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

- The processing time P_i are uniformly distributed in the interval [1,10].
- The due dates d_i are 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 0.2,0.4,0.6,0.8and 1.0, are considered . For each selected value of *n*, one problem was generated for each of five values of parameters producing five problems for each value of *n* . The BAB and B algorithms were tested by coding them in matlab7 and running on Pentium IV at 2800MHz with Ram 512MB computer. The BAB algorithm is tested on problems with size (10,20,30) .

For problems that are not solved to optimally because the execution time exceed 30 minutes, the optimal solution for these unsolved problems found by our algorithm B .

Table(1) shows the results for problem (P4) obtained by BAB algorithm. The first column "n" refers to the number of jobs, the second column "EX" refers to the number of example for each instance n, the third column "optimal" refers to the optimal value obtained by BAB algorithm for problem (P4), the fourth column "UB" refers to the upper bound , the fifth column "ILB" refers to the initial lower bound , the sixth column "nodes" refers to the number of nodes , the seventh column "time" refers to the time cost 'by seconds' to solve the problem, the last column "status" refers to the problem solved '0' or not '1'. The symbol "*" refers to the optimal=UB, we stopped when the sum of status' column $\geq$3.

Table(2) , show the results for problem (P4) obtained by algorithm (B). The first two columns as the same columns in table(1), the third column "value" refers to the minimum value that we get by algorithms B, and the last column "time" refers to the time cost 'by seconds' to solve the problem .

Table (3)compare between *BAB and algorithm (B) to solve a problem(P4)(time by seconds).* It is clear from table (3) that the BAB method can not solved problems with n$\geq$30.

Table -1:*The performance of initial lower bound,upper bound and computational time of BAB algorithm for (P4).*

n	EX	optimal	UB	ILB	nodes	time	status
10	1	5868	5868*	5863	222	0.083753	0
	2	14653	14653*	14644	668	0.05233	0
	3	9629	9629*	9614	241	0.022428	0
	4	9798	9803	9792	71	0.008206	0
	5	5518	5518*	5499	266	0.025065	0
20	1	74030	74030*	74010	17887	1.331111	0
	2	52867	52877	52810	417071	29.22711	0
	3	78694	78694*	78661	1043412	81.99179	0
	4	71171	71182	71096	240860	16.87217	0
	5	46104	46114	46037	1097334	78.07866	0
30	1	210893	210893*	210891	23445581	1800	1
	2	157867	157867*	157847	22422037	1800.001	1
	3	157309	157318	157293	21096053	1800.001	1
	4	215258	215268	215137	21573266	1800.001	1
	5	232131	232131*	232053	21280420	1800	1

Table -3:*BAB Vs algorithm (B) to solve a problem(P4) (time by seconds).*

n	BAB	B
10	0.02	0.01
20	40	0.09
30	1800	0.1
100	1800	0.1
200	1800	75
300	1800	200
400	1800	510
500	1800	1550

Table - 2:*Results of algorithm B for (P4).*

n	ex	value	time
10	1	5868	0.09719246
	2	14653	0.00887009
	3	9629	0.01406612
	4	9798	0.01021048
	5	5518	0.0208633
20	1	74030	0.08547378
	2	52867	0.13290581
	3	78694	0.06773622
	4	71171	0.16499083
	5	46104	0.15121613
30	1	210893	0.06058998
	2	157867	0.08748179
	3	157309	0.09807895
	4	215258	0.46199023
	5	232131	0.26577637
100	1	5868	0.05231801
	2	14653	0.010738
	3	9629	0.01390555
	4	9798	0.01086037
	5	5518	0.02007836
200	1	46373355	71.2329712
	2	55521683	74.9988027
	3	52408900	114.055689
	4	43625257	47.1273153
	5	45305234	78.6391166
300	1	180481870	28.6847228
	2	189083219	252.168852
	3	183437469	127.6062
	4	157888207	223.006137
	5	164142954	352.721073
400	1	421801412	308.649517
	2	423348248	848.978298
	3	430915881	921.210297
	4	394605521	355.380809
	5	413688655	531.261101
500	1	712946851	1496.9584
	2	763797269	1551.28829
	3	755495302	1676.87384
	4	782832453	1768.17998
	5	750245041	1633.0393

REFERENCES

1) Azizoglu M.,Kondakci S., and Kokslan M.," Single machine scheduling with maximum earliness and number tardy", Computers & Industrial Engineering ,45 : 257–268 (2003).
2) Bagga P.C., Kalra K.R.," A node elimination procedure for Townsend's algorithm for solving the single machine quadratic penalty function scheduling problem", Management Science, 26, :633–636(1980).
3) Chang P., Su L.," Scheduling n jobs on one machine to minimize the maximum lateness with a minimum number of tardy jobs", computer& industrial engineering, 40 : 49-60(2001).
4) Chen T., Qi X., and Tu F., "Single machine scheduling to minimize weighted earliness subject to maximum tardiness", computer of operation research, 24:147-152(1997).
5) Cheng ,et al.," A survey of scheduling problems with setup times or costs", European Journal of Operational Research 187: 985–1032 (2008).
6) Emmons H., "A note on a scheduling problem with dual criteria", Naval Res. Logis. Quarterly, 22: 615-516 (1975).
7) French S.," Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop", John Wiley & Sons, New York(1982).
8)George S., Paul S., " Pareto optima for total weighted completion time and maximum lateness on a single machine", Discrete Applied Mathematics 155: 2341 – 2354(2007).
9)Graham R.L., Lawler E.L., Lenstra J.K., and Rinnooy K.A.," Optimization and approximation in deterministic sequencing and scheduling: A survey", Annals of Discrete Mathematics 5: 287–326(1979).
10) Gupta S. K., Sen T.," On the single machine scheduling problem with quadratic penalty function of completion times: an improved branching procedure", Management Science, 30: 644–647(1984).
11) Hoogeveen J.A, " Invited Review Multi-criteria scheduling ", European Journal of Operational Research 167 : 592–623(2005).
12) Hoogeveen, J.A., van de Velde, S.L., " Minimizing total completion time and maximum cost simultaneously is solvable in polynomial time", Operations Research Letters 17: 205–208(1995).
13) Jackson J.R.," Scheduling a production line to minimize maximum tardiness", Research Report 43, Management Sciences Research Project, UCLA(1955).
14) Jouni L., "Multi-objective Nonlinear Pareto-Optimization ", Lappeenranta University of Technology(2000).

15) Kawi Haidar Y. ,”Single machine scheduling for multi-objective functions”, M.Sc. thesis , Dept. of mathematics, college of science, University of AL-Mustansiriya,(2009).
16) Lawler E.L.,"Optimal sequencing of a single machine subject to precedence constraints"Management Science 19/5:544-546 (1973).
17) Lee C.Y., Vairaktarakis G.L.,” Complexity of single machine hierarchical scheduling: A survey”, Complexity in Numerical Optimization, 19: 269–298(1993).
18) Lenstra J.K.,Rinnooy Kan, and Brucker P.,"Complexity of machine scheduling problems", Annals of Discrete Mathematics 1, :343-362(1979).
19) Moslehi G., Moghaddam R., Vasei M., and Azaron A., "A branch-and-bound algorithm for a single machine sequencing to minimize the sum of maximum earliness and tardiness with idle insert ",Applied Mathematics and Computation 174 : 388–408(2006)
20) Nagar A., Jorge H., and Sunderesh H., " Multiple and bi-criteria scheduling: A literature survey", European Journal of Operational Research North-Holland,81:8-104 (1995).
21) Nelson R.T., Sarin R.K., and Daniels R.L., "Scheduling with multiple performance measures: The one-machine case", Management Science 32: 464-479(1986).
22) Smith W.E. , "Various optimizers for single stage production", Naval Research Logistics Quarterly 3/1: 59-66(1956).
23) Szwarc W., Posner M. E., and Liu J.,” The single machine problem with quadratic cost function of completion times”, Management Science, 34: 1480–1488(1988).
24) Townsend W. ,” The single machine problem with quadratic penalty function of completion times: a branch and bound solution”, Management Science, 24: 530–534(1978).
25) Van Wassenhove L.N., and Gelders F., "Solving a bicriterion scheduling problem", European Journal of Operational Research 4/1: 42-48(1980).