Branch and Bound Method to Minimized Three Criteria

Mohammed K.Al-Zuwaini
Department of Mathematics,College of computer science and Mathematics, Thi-Qar University
mkzz50@yahoo.com
Kadhem M. Hashem
Department of computer, College of Education for Pure sciences, Thi-Qar University
Kadhem1962@yahoo.com
Jafar S.Aneed
Department of Mathematics, College of Education for Pure sciences, Thi-Qar University
J_S27@yahoo.com

Abstract
This paper addresses the problem of minimizing the sum of total completion times, maximum earliness and maximum tardiness on a single machine with unequal release date. A branch and bound algorithm with forward approach, in order to find the exact (optimal) solution for itwith two lower bounds (LB_1, LB_2) and four upper bounds(UB_1, UB_2, UB_3, UB_4) that introduced in this paper. Ten special cases are suggested and proved that yield optimal solution. In general, this problem is strongly NP-hard, and solved it with up to 30 jobs.

Keywords: Single machine scheduling, multi-criteria, simultaneous, release date, total completion times, maximum earliness and maximum tardiness**.**

الملخص :

في هذا البحث تم اعتماد مسألة تزغير المجموع لوقت الاكتمال وتكبير وقت التبكير وتكبير وقت التأخير لمسألة الجدوله لماكنه واحدة وفي اوقات تسليم غير متساويه . أستخدمة طريقة التفرع والتقيد الاماميه لايجاد وقت الحل الامثل لهذه المسألة وبحدين للقيد الادنى LB_1, LB_2 وبأربعة قيود عليا (UB_1, UB_2, UB_3, UB_4) . تم تطبيق الحلول على عشرة حالات خاصه في ايجاد الحل الامثل بشكل عام هذه المسألة هي NP-hard قويه والحل يمكن الحصول عليه لغاية 30 عمل .

1. Introduction

In general, multi-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] prove 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)[23] 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[10]; Nelson et al., 1986[19]; George S., and Paul S. 2007[11]), 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 [16], [18] review the problem in its general form whereas Lee and Vairaktarakis (1993)[17] 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)[12] studies a number of bi-criteria scheduling problems. Also, there are some papers about this object (Cheng et al. 2008[9], George S., et al.2007[11], and Azizoglu et al. 2003 [5]).

In this paper the problem of scheduling n independent jobs on a single machine is considered to minimize (Multi-criteria Objective Function (MOF)), the sum of total completion time, maximum earliness and maximum tardiness by using the (BAB) method. This problem is denoted by $1/r_i/\sum C_i+E_{max}+T_{max}$.

2. Problem Formulation

Single machine scheduling models seem to be very simple but are very important for understanding and modeling multiple machines models. A set N={1,2,…,n} of n independent jobs has to be scheduled on a single machine in order to optimize a given criterion. This study concerns the one machine scheduling problem with multiple objectives function denoted by $(1/r_i/\sum_{i=1}^{n} C_i +E_{max}+T_{max})$. In this problem, preemption is not allowed, no precedence relation among jobs is assumed, only one job i can be processed at a time. Each job i has a release date r_i at which it cannot be processed before, needs p_i time units to be processed on the machine, and ideally should be completion at its due date d_i. For each job i calculated the slack times$_i=d_i - p_i$. The objective is to find a schedule to minimize the sum of total ofcompletion times ($\sum_{i=1}^{n} C_i$), maximum earliness (E_{max}) and maximum tardiness(T_{max}).

The problem $(1/r_i/\sum_{i=1}^{n} C_i +E_{max}+T_{max})$ can be stated as follows:

A set of n independent jobs N={1,2,…,n} are available for processing at time r_i, job i (i=1,2,…,n) is to be processed with uninterruption on a single machine that can be handle only one job at a time, requires processing time p_i, and ideally should be completed at its due date d_i. For a given sequence π of the jobs, completion time of job i, $C_{\pi(i)}$, earliness $E_{\pi(i)}$, and the tardiness $T_{\pi(i)}$ are given by:

$$C_{\pi(1)} = r_{\pi(1)} + p_{\pi(1)}$$

$$C_{\pi(i)} = \max\{C_{\pi(i-1)}, r_{\pi(i)}\} + p_{\pi(i)} \qquad i = 2, \dots, n$$

$E_{\pi(i)}=\max\{d_{\pi(i)} - C_{\pi(i)}, 0\}$ and $T_{\pi(i)}=\max\{C_{\pi(i)} - d_{\pi(i)}, 0\}$, i=1,..,n

The problem $1/r_i/T_{max}$ is strongly NP-hard [20] and $1/r_i/\sum_{i=1}^{n} C_i$is strongly NP-hard [15] and the problem $1//\sum_{i=1}^{n} C_i +E_{max}$ is NP-hard , [1,3,14].

The aim is to find a sequence π that minimizes the total cost $R=\sum_{i=1}^{n} C_{\pi(i)} + E_{max}(\pi) + T_{max}(\pi)$. The mathematic form of this problem which denoted by (S)can be stated as follows:

$$\left.\begin{array}{l} \text{Min } R = \text{Min}_{\pi\in\delta}\{\sum_{i=1}^{n} C_{\pi(i)} + E_{max}(\pi) + T_{max}(\pi)\} \\ \text{s.t.} \\ \left.\begin{array}{ll} C_{\pi(i)} \geq r_{\pi(i)} + p_{\pi(i)} & i = 1,2,\dots,n \\ C_{\pi(i)} \geq C_{\pi(i-1)} + p_{\pi(i)} & i = 2,3,\dots,n \end{array}\right\}\cdots(1) \\ E_{\pi(i)} \geq d_{\pi(i)} - C_{\pi(i)} \qquad i = 1,2,\dots,n \\ T_{\pi(i)} \geq C_{\pi(i)} - d_{\pi(i)} \qquad i = 1,2,\dots,n \\ E_{\pi(i)} \geq 0, T_{\pi(i)} \geq 0, r_{\pi(i)} \geq 0, p_{\pi(i)} > 0 \qquad i = 1,2,\dots,n \end{array}\right\}\dots.(S)$$

where $\pi(i)$ denotes the position of job i in the ordering π and δ denotes the set of all enumerated schedules.

3. Decomposition of Problem (S)

In this section the problem (S) is decomposed into three subproblems with a simple structure. Some results are stated which help in solving the problem (S).

The problemS can be decomposed into three subproblems say (SA_1), (SA_2) and $(SA_3)$where :

$$\left.\begin{array}{l} N_1 = \min_{\pi\in\delta}\{\sum_{i=1}^{n} C_{\pi(i)}\} \\ \text{S.t} \\ \qquad (1) \\ r_{\pi(i)} \geq 0, p_{\pi(i)} > 0 \qquad i = 1,2,\dots,n \end{array}\right\}\cdots(SA_1)$$

$$\left.\begin{array}{l} N_2 = \min_{\pi\in\delta}\{E_{max}(\pi)\} \\ \text{S.t} \\ \qquad (1) \\ E_{\pi(i)} \geq d_{\pi(i)} - C_{\pi(i)} \qquad i = 1,2,\dots,n \\ E_{\pi(i)} \geq 0, r_{\pi(i)} \geq 0, p_{\pi(i)} > 0 \qquad i = 1,2,\dots,n \end{array}\right\}\cdots(SA_2)$$

$$\left.\begin{array}{l} N_3 = \min_{\pi\in\delta}\{T_{max}(\pi)\} \\ \text{S.t} \\ \qquad (1) \\ T_{\pi(i)} \geq C_{\pi(i)} - d_{\pi(i)} \qquad i = 1,2,\dots,n \\ T_{\pi(i)} \geq 0, r_{\pi(i)} \geq 0, p_{\pi(i)} > 0 \qquad i = 1,2,\dots,n \end{array}\right\}\cdots(SA_3)$$

Theorem (1)[4]

$N_1 + N_2 + N_3 \leq R$ where N_1 ,N_2 ,N_3 and R are the minimum objective function values of (SA_1), (SA_2), (SA_3) and S respectively.

4. Special Cases

A machine scheduling problem of type NP-hard is not easily solvable and it is more difficult when the objective function is multi objective. Using some mathematical programming techniques to find the optimal solution for this kind of problem as: dynamic programming and branch and bound method. Sometimes special cases for this problem can be solved. A special case for scheduling problem means finding an optimal schedule directly without using mathematical programming techniques. A special case if it exists depends on satisfying some conditions in order to make the problem easily solvable. These conditions depend on the objective function as well as the jobs [13]. In this section some special cases of problem (S) are given.

Case(1): The SRD rule gives an optimal solution for problem (S) ifp_i=p and$d_i = ip$for all i in SRD.

Proof:

Since $d_i = ip \forall$ i in SRD, then E_{max} = 0 and $T_{max} = \sum_{i=1}^{n} I_i$. Then the problem (S) reduced to $1/r_i, p_i = p/\sum_{i=1}^{n} C_i + T_{max}$.

Now, since p_i=pfor all i in SRD, then $\sum_{i=1}^{n} C_i = \sum_{i=1}^{n} \sum_{j=1}^{i} I_j + (\frac{n^2+n}{2})p$. But $(\frac{n^2+n}{2})p$ is constant, then $\sum_{i=1}^{n} C_i = \sum_{i=1}^{n} \sum_{j=1}^{i} I_j$(i.e. a schedule that is optimal solution with respect to $\sum_{i=1}^{n} C_i$ is also optimal with respect to$\sum_{i=1}^{n} \sum_{j=1}^{i} I_j$).But $\sum_{i=1}^{n} \sum_{j=1}^{i} I_j \cong C_{max}$ [21].

Carliar, (1982)[8] show that SRD schedule is optimal schedule for C_{max} .

Hence SRD rule gives an optimal solution for problem (S).

Case(2): If $p_1 \leq p_2 \leq \ldots \leq p_n$,$r_1 \leq r_2 \leq \ldots \leq r_n$ and $C_i = d_i \forall$ i in a scheduleSPT, then SPT is an optimal solution for problem (S).

Proof:

Since $C_i = d_i \forall$ i in SPT, then $E_{max} = T_{max}$ =0.The problem (S) reduced to $1/r_i / \sum_{i=1}^{n} C_i$ **,** but this problem solved in SPT rule [2].

Case(3): IfC_i=$d_i \forall$ iin a schedule πand the preemptive is allowed, then πgives an optimalsolutionfor the problem $1/r_i, pmtn/\sum_{i=1}^{n} C_i + E_{max} + T_{max}$.

Proof:

SinceE_{max} =T_{max} =0 in π, then the problem (S) reduced to $1/r_i, pmtn/\sum_{i=1}^{n} C_i$, but [6] solved this problem by (SRPT) rule. Then π gives an optimal solution for the problem $1/r_i, pmtn/\sum_{i=1}^{n} C_i + E_{max} + T_{max}$ provided that $C_i = d_i \forall i \in \pi$.

Case(4): If in SPT scheduler$_i$ =r$\forall$ iand satisfy Just In Time(JIT),then SPT gives an optimal solution for the problem $1/r_i = r/\sum_{i=1}^{n} C_i + E_{max} + T_{max}$.

Proof:

From (JIT) we get$E_{max}=T_{max}=0$, then the problem (S) reduced to $1/r_i=r/\sum_{i=1}^{n}C_i$. But this problem was solved by (SPT) rule. Then SPT gives an optimal solution for the problem $1/r_i=r/\sum_{i=1}^{n}C_i+E_{max}+T_{max}$.

Case(5): Any schedule gives anoptimal solution for the problem (S), if$r_i=r,p_i=p$ and $d_i=d\ \forall\ i$ (i=1,2,...,n).

Proof:

Since $\sum_{i=1}^{n}C_i=nr+(\frac{n^2+n}{2})p$, $E_{max}=\max\{d-(r+p),0\}$ and$T_{max}=\max\{(r+np)-d,0\}$ in any schedule. Then any schedule is optimal for the problem $1/r_i=r,p_i=p,d_i=d/\sum_{i=1}^{n}C_i+E_{max}+T_{max}$ (because the three quantities are constant).

Case(6): The EDD schedule gives an optimal solution for the problem (S), if $r_i=r,p_i=p\forall\ i$ in EDD and $E_{max}(EDD)=E_{max}(MST)$.

Proof:

Since any sequence gives an optimal solution for $1/r_i=r,p_i=p/\sum_{i=1}^{n}C_i$ problem and EDD rule gives an optimal solution for $1/r_i=r/T_{max}$ problemand MST gives an optimal solution for $1/r_i=r/E_{max}$ problem. But $E_{max}(EDD)=E_{max}$ (MST). So EDD gives an optimal solution for $1/r_i=r,p_i=p/\sum_{i=1}^{n}C_i+E_{max}+T_{max}$ provided that $E_{max}(EDD)=E_{max}(MST)$.

Case(7): The MST schedule is optimal solution for the problem (S), if $r_i=r,p_i=p\forall\ i$ in MST and T_{max} (MST)=T_{max} (EDD).

Proof:

Since any sequence gives an optimal solution for $1/r_i=r,p_i=p/\sum_{i=1}^{n}C_i$ problem and MST rule gives an optimal solution for $1/r_i=r/E_{max}$ problem and EDD gives an optimal solution for $1/r_i=r/T_{max}$ problem. But $T_{max}(MST)=T_{max}$ (EDD). So MST gives an optimal solution for $1/r_i=r,p_i=p/\sum_{i=1}^{n}C_i+E_{max}+T_{max}$ provided that $T_{max}(MST)=T_{max}$ (EDD).

Case(8): If $r_i=r,d_i=d\forall\ i(i=1,2,\ldots,n)$ and $\sum_{i=1}^{n}C_i$ (MST)=$\sum_{i=1}^{n}C_i$ (SPT),then MST schedule is optimal solution for the problem $1/r_i=r,p_i=p/\sum_{i=1}^{n}C_i+E_{max}+T_{max}$.

Proof:

From conditions $r_i=r,d_i=d\forall\ i(i=1,2,\ldots,n)$ any order gives optimal solution for problem $1/r_i=r,d_i=d/T_{max}$. Now, since $\sum_{i=1}^{n}C_i$ (MST)=$\sum_{i=1}^{n}C_i$ (SPT), then MST minimum the problem $/r_i=r,d_i=d/\sum_{i=1}^{n}C_i+E_{max}+T_{max}$.

Case(9): If $r_i=r,d_i=d\forall\ i(i=1,2,\ldots,n)$ and $E_{max}(SPT)=E_{max}(MST)$, then SPT schedule is optimal solution for the problem $1/r_i=r,d_i=d/\sum_{i=1}^{n}C_i+E_{max}+T_{max}$.

Proof:

From conditions $r_i=r,d_i=d\forall\ i(i=1,2,\ldots,n)$ any order gives optimal solution for problem $1/r_i=r,d_i=d/T_{max}$. Now, since $E_{max}(SPT)=E_{max}(MST)$, then SPT minimize the problem $/r_i=r,p_i=p/\sum_{i=1}^{n}C_i+E_{max}+T_{max}$.

Case(10): If $r_i=r$ and (SPT) schedule gives $d_i+p_j\leq d_j\forall\ i\prec j$ and T_{max} (SPT)=T_{max} (EDD), then SPT is optimal solution for the problem (S).

Proof:

Since $d_i + p_j \leq d_j \forall\ i \prec j$ in SPT schedule then $d_i - p_i \leq d_j - p_j \forall i \prec j$ ($p_i \geq 0$). Thus SPT gives optimal solution for both criteria E_{max} and $\sum C_i$, and from condition T_{max} (SPT)=T_{max} (EDD). Then SPT is optimal solution for the problem (S).

5. Upper Bound (UB)

In this section, use four heuristic methods for ordering the jobs and evaluate the cost of problem (S).

Heuristic (1): Order the jobs according to (SPT) rule, and find $UB_1=\sum_{i=1}^{n} C_i+E_{max}+T_{max}$ (SPT).
Heuristic (2): Order the jobs according to (MST) rule, and find $UB_2=\sum_{i=1}^{n} C_i+E_{max}+T_{max}$ (MST).
Heuristic (3): Order the jobs according to (EDD) rule, and find $UB_3=\sum_{i=1}^{n} C_i+E_{max}+T_{max}$ (EDD).
Heuristic (4): Order the jobs according to (SRD) rule, and find $UB_4=\sum_{i=1}^{n} C_i+E_{max}+T_{max}$ (SRD).
The heuristic which gives a minimum cost of the problem (S) among these heuristics is chosen to be an upper bound, (i.e. UB=min{ UB_1, UB_2, UB_3, UB_4}). This UB is then used in a root node of the search tree in a branch and bound method.

Example(1): The upper bound illustrate in four jobs scheduling problems

r_i
p_i
d_i 2 l)

Solution:

The SPT schedule is (2,1,4,3) then $UB_1=\sum_{i=1}^{4} C_i+E_{max}+T_{max}$ =64.

The MST schedule is (1,3,4,2) then $UB_2=\sum_{i=1}^{4} C_i+E_{max}+T_{max}$ =55.

The EDD schedule is (1,4,3,2) then $UB_3=\sum_{i=1}^{4} C_i+E_{max}+T_{max}$ =58.

The SRD schedule is (1,2,3,4) then $UB_4=\sum_{i=1}^{4} C_i+E_{max}+T_{max}$ =52.

Hence UB=min{ UB_1, UB_2, UB_3, UB_4}=52.

It should be noted that an optimal sequence is (1,2,4,3) for this example, and the optimal value is 50 which is obtained by using complete enumeration.

6. Lower Bound (LB)

Deriving a lower bound for a problem (S) that has a multiple objective function is very difficult since it is not easy to find the minimum cost for the three objectives. Since the problem (S) is strongly NP-hard may be find a lower bound that gives minimum value for one of them but not all.

In this section two lower bounds LB_1 and LB_2 are derived for problem (S).

6.1. The First Lower Bound (LB_1)

The first lower bound is based on decomposing (S) into three subproblems (SA_1), (SA_2) and (SA_3) as shown in Section (3), then N_1 was calculated to be the lower bound for (SA_1) by (SRPT) rule (sequencing the jobs in non-decreasing order of Shortest Remaining Processing Time)[6], N_2 was calculated to be the lower bound for (SA_2) by (MRST) rule (sequencing the jobs in non-decreasing order of Minimum Remaining Slack Time)[7], N_3 was calculated to be the lower bound for (SA_3) by (MEDD) rule (sequencing the jobs in non-decreasing order of Smallest Remaining Due Date)[7] and then applying Theorem(1) to get the first a lower bound for problem (S).

Example(2): The first lower bound was illustrate in four jobs scheduling problems

r_i

p_i

d_i 2 l)

Solution:

For the relax problem $1/r_i, pmtn/\sum_{i=1}^{n} C_i$, the (SRPT) rule shown in the **Fig** (1a) below

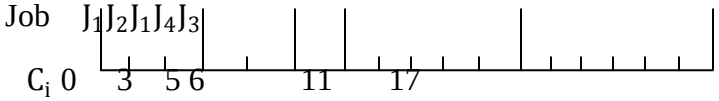

Fig: (1a)

$C_1=6$ $C_2=5$ $C_3=17$ $C_4=11$

$N_1=\sum_{i=1}^{4} C_i=39.$

For the relax problem $1/r_i, pmtn/E_{max}$, the (MRST) rule shown in the **Fig** (1b) below

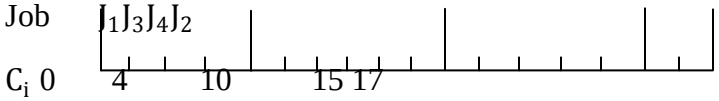

Fig: (1b)

$C_1=4$ $C_2=17$ $C_3=10$ $C_4=15$

$E_1=4$ $E_2=0$ $E_3=1$ $E_4=0$

$N_2=E_{max}=4.$

For the relax problem $1/r_i, pmtn/T_{max}$, the (MEDD) rule shown in the **Fig** (1C) below

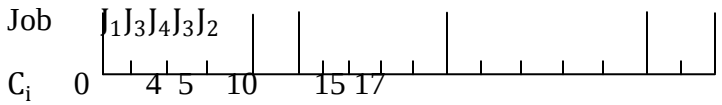

Fig: (1C)

$C_1=4$ $C_2=17$ $C_3=15$ $C_4=10$

$T_1=0$ $T_2=5$ $T_3=4$ $T_4=0$

$N_3=T_{max}=5$.

Then $LB_1=N_1+N_2+N_3=39+4+5=48$.

6.2 The second Lower Bound (LB_2):

The second lower bound can calculated for the problem (S) by using the relaxation of constraints of objective function as follows:

For the problems (SA_1) and (SA_3), we assume that $r^*=\min_{1\le i\le n}\{r_i\}$and $r_i = r^*$,i=1,2,…,n, to get the problems $1/r_i = r^*/\sum_{i=1}^{n} C_i$ and$1/r_i = r^*/T_{max}$,which are solved by SPT and EDD rules respectively. For the problem (SA_2), we assume that $r^\circ=\max_{1\le i\le n}\{r_i\}$and $r^\circ=r_i$,i=1,2,…,n, to get the problem $1/r_i = r^\circ/E_{max}$, which are solved by MST rule and then applying Theorem (1) to get the second lower bound for problem (S).
Hence the lower bound is $LB=\max\{LB_1, LB_2\}$.

Example(3): The first lower bound illustrate in four jobs scheduling problems

r_i

p_i

d_i 2 l)

Solution:

Let $r^*=\min_{1\le i\le 4}\{r_i\}=0$, then SPT give the schedule (2,1,4,3) with $N_1 = \sum_{i=1}^{4} C_i=36$ and the EDD give the schedule (1,4,3,2) with $N_3=T_{max}=5$.

Let $r^\circ=\max_{1\le i\le 4}\{r_i\}=2$, then MST give the schedule (1,3,4,2) with $N_2=E_{max}=2$.

Then $LB_2=N_1+N_2+N_3=36+2+5=43$.

7. Branch and Bound (BAB) Algorithm [4]

In this section, a description of branch and bound (BAB) algorithm is given and its implementation. The heuristic methodis applied at the top of search tree (root node) to provide an upper bound (UB) on cost of an optimal schedule is obtained by choosing the upper bound from

Section (5). Also at the top of the search tree an initial lower bound (ILB) on the cost of an optimal schedule is obtained by choosing the better of two lower bounds from Section (6.2). The algorithm uses a forward sequencing branching rule for which nodes at level k of the search tree correspond to initial sequences in which jobs are sequenced in the first k positions. The branching procedure describes the method to partition a subset of possible solution. These subsets can be treated as a set of solutions of corresponding subproblems of the original problem. The bounding procedure indicates how to calculate a lower bound (LB) on the optimal solution value for each subproblem generated in the branching process. The search strategy describes the method of choosing a node of the search tree to branch from it; we usually branch from a node with smallest lower bound (LB) among the recently created nodes.

8. Computational Experience

An intensive work of numerical experimentations has been performed.
Subsection (8.1) shows how instances (test problems) can be randomlygenerated.

8.1. Test Problems

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

1- The processing time p_i is uniformly distributed in the interval [1, 10].

2- The release date $r_i$is uniformly distributed in the interval [0, αP], where $\alpha \in [0.125, 0.25, 0.50, 0.75, 1.00]$ and $P = \sum_{i=1}^{n} p_i$.

3- 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, five problemswere generated.

8.2. Computational Experience with the Lower and UpperBounds of (BAB) Algorithm

The (BAB) algorithm was tested by coding it in MATLAB 7.10.0.499 (R2010a) and implemented on Intel(R) Core(TM)i3 CPU M380 @ 2.53 GHZ, with RAM 3.00 GB(2.87GB usable) personal computer.
Table (1) below, shows the results for problem (S) obtained by (BAB) algorithm. The first column "n" refers to the number of jobs, the second column "EX" refers to the number of examples for each instance n, where $n \in \{5,10,15,20,25,30\}$, the third column "Optimal" refers to the optimal values obtained by (BAB) algorithm for problem (S), 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 second' to solve the problem, the last column "Status" refers to the problem solved '0' or not solved '1'. The symbols

"*" refers to the UB gives an optimal solution and "**" refers to the ILB gives an optimal solution. The (BAB) algorithm was stopped when the sum of "status column≥ 3".

A condition for stopping the (BAB) algorithm was determined and considering that the problem is unsolved (state is 1), that the (BAB) algorithm is stopped after a fixed period of time, here (after 30 minutes).

If the value of UB=ILB then the optimal is UB and there is no need to branch the search tree of (BAB) algorithm.

Table (1): The performance of initial lower bound, upper bound, number of nodes and computational time in second of (BAB) algorithmfor (S).

N	EX	Optimal	UB	ILB	Nodes	Time	Status
5	1	127	127*	125	36	0.00673	0
	2	57	58	57**	14	0.0035	0
	3	87	93	87**	22	0.0080	0
	4	77	77*	76	14	0.0025	0
	5	60	60*	60**	0	0.0008	0
10	1	352	401	336	1396	0.1488	0
	2	201	226	196	1547	0.1583	0
	3	337	368	327	414	0.0449	0
	4	389	408	376	1265	0.1385	0
	5	146	195	143	67	0.0128	0
15	1	660	744	644	12529	1.5424	0
	2	557	672	549	16211	2.3152	0
	3	720	803	720**	210	0.0337	0
	4	353	378	331	273	0.0473	0
	5	567	599	532	252493	35.0969	0
20	1	953	1030	935	2508609	264.9413	0
	2	1091	1281	1064	2508609	71.6318	0
	3	789	912	762	40143	3.9402	0
	4	981	1144	913	95043	9.4424	0
	5	824	929	769	5501	0.5584	0
25	1	1502	1908	1436	4230692	434.4436	0
	2	1649	2021	1570	3974179	398.7797	0
	3	1877	2040	1746	17151377	1800.0001	1
	4	1363	1547	1359	487324	51.7407	0
	5	1847	1963	1757	17791999	1800.0011	1
30	1	2883	3278	2767	16378467	1800.0003	1
	2	2106	2578	2076	5231216	568.0532	0
	3	1973	2420	1878	16315296	1800.0001	1
	4	1873	2418	1796	4174669	470.0744	0
	5	2086	2599	2056	665806	73.6937	0

Table (2) summarizes Table (1)

Table (2): Summary of Table (1) of (BAB) algorithm

N	Av.Nodes	Av.Time	Unsolved problem
5	17.2	0.164	0
10	937.8000	0.1007	0
15	5.6343	7.8071	0
20	1031581	70.1028	0
25	8.7271	294.9880	2
30	8.5531	370.6000	2

Table (2) is the summary of Table (1), and shows the average ofnodes and computational times for the solved problems. It also shows theunsolved problems among the 5 problems of each n, where $n \in \{5,10,15,20,25,30\}$.

Conclusions

In this paper, the problems of scheduling jobs on one machine for a variety of three-criteria are considered. Branch and bound algorithm is proposed to find exact (optimal) solution for the problem $1/r_i/\sum_{i=1}^{n} C_i + E_{max} + T_{max}$ with two lower bounds (LB_1, LB_2), an upper bound UB. Also, ten special cases are derived and proved for the last problem. It is hoped that the contribution of this paper would provide an incentive increased research effort in this multi-criteria field especially three criteria.

References

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

[2] AL. Zuwaini M. K., " Single Machine Scheduling to Minimize Total Cost of Flow Time With unequale Ready Times., M. Sc. thesis, Dept. of Mathematics, College of Education, Basrah University (2000).

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

[4] Araibi S.M.," Machine Scheduling Problem to Minimize Two and Three Objectives Function"., M.Sc. thesis, Dept. of Mathematical, College of Education for Pure Sciences, Thi-Qar University(2012).

[5] Azizoglu M.,Kondakci S., and Kokslan M.," Single machine scheduling with maximum earliness and number tardy", Computers & Industrial Engineering ,45 , 257–268 (2003).

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

[7] Baker K.R., and Z. Su, "Sequencing with due-dates and early start times to minimize maximum tardiness," Naval Research Logistics, vol. 21, no. 1, pp. 171-176,1974.

[8] Carliar, J. "The one-machine sequencing problem". European J. of operational Research 11(1982) pp.42-47.

[9] Cheng ,et al.," A survey of scheduling problems with setup times or costs", European Journal of Operational Research 187, 985–1032 (2008).

[10] French S.," Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop", John Wiley & Sons, New York(1982).

[11] 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).

[12] Hoogeveen J.A, " Invited Review Multi-criteria scheduling ", European Journal of Operational Research 167 , 592–623(2005).

[13] Husein N.A., " Machine Scheduling Problem to Minimize Objective Function ", M.Sc. thesis, Dept of Mathematical, College of Education (Ibn AL Haitham), Baghdad University(2012).

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

[15] Lawler E.L., J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys , "Sequencing and Scheduling: Algorithms and Complexity" in S.C. Graves, A.H.G. Rinnooy Kan and P. Zipkin (Eds.): Handbooks in Operations Research and Management Science vol 4: Logistics of Production and inventory, North-Holland, Amsterdam (1993).

[16] Lenstra J.K.,Rinnooy Kan, and Brucker P.,"Complexity of machine scheduling problems", Annals of Discrete Mathematics 1, 343-362(1979).

[17] Lee C.Y., Vairaktarakis G.L.," Complexity of single machine hierarchical scheduling: A survey", Complexity in Numerical Optimization, 19, 269–298(1993).

[18] Nagar A., Jorge H., and Sunderesh H., " Multiple and bi-criteria scheduling: A literature survey", European Journal of Operational Research North-Holland,81,88-104 (1995).

[19] 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).

[20] Pinedo, M.L.,"Scheduling theory, algorithms, and systems", Springer Science+Business Media, LLC., New York (2008).

[21] Rinnooy Kan, A.H.G. "Machine scheduling problem: classification, complexity and computations, Martinus Mijhoff, the Mague. Holl and (1976).

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

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