Algorithms for Multicriteria Scheduling Problems

Tariq S. Abdul-Razaq and Karar F. Abdul-Razaq
Math. Dept, College of Sciences, University of Al-Mustansiriya, Baghdad, Iraq.

Abstract:

In this paper, we consider the multicriteria scheduling problem on single machine to minimize two criteria: maximum cost function, denoted by maximum late work (V_{max}) and maximum earliness (E_{max}). We propose several algorithms based on types of objectives function to be optimized. The solutions of the proposed procedures are compared with that of the optimal solutions and Pareto optimal solutions for the smaller instance size, these algorithms dealing with hierarchical minimization problem as well as simultaneous minimization problem with and without weight. Computational results show the usefulness of these procedures.

Keywords: *Scheduling, single machine, hierarchical, simultaneous minimization, algorithms, Branch and Bound.*

1. Introduction

The bicriteria scheduling problem has received significant attention in recent years [1]. The basic of bicriteria can be stated as follows. There are n jobs to be processed on a single machine, each job i has processing time p_i and due date d_i at which ideally should be completed. Penalties are incurred whenever a job is completed earlier or later than it's due date. So the problem is a bicriteria scheduling problem.

Let Ω be a schedule, E(Ω),V(Ω) be functions of earliness and late work respectively.

The problem is to find a schedule Ω to optimize E(Ω),V(Ω) or a composite objective function of E(Ω) and V(Ω).

In the literature [3], there are mainly three classes of approaches that applicable to bicriteria scheduling problem.

C1: The hierarchical approach, one of the criteria (more important) regard as constraint (primary) criterion which must be satisfied and the other one considered as (secondary) criterion to optimize.

This means optimize the primary criterion while breaking ties in favor of the schedule that has minimum secondary criterion [2].

C2: Minimizing a weighted sum of the bicriteria (objectives) and convert the bicriteria to a single criterion problem, several bicriteria scheduling problems studied belong to this class [4],[9].

C3: One typically generates all efficient (Pareto optimal) schedules and select the one that yields the best composite objective function value of two criteria. For the bicriteria that concern the simultaneously minimization of (Σc_i ,fmax) for 1//F(Σc_i ,fmax) problem in C3 which is solved [4] in a polynomial time [11] solved the 1//F(Σc_i ,Tmax) problem.

In this paper we will study bicriteria problems belong to the classes C1, C2 and C3. In section (2), notation, basic concepts and mathematical forms are given. In sections (3) and (4) we formulate the multicriteria problem according to the classes of the approaches, we propose algorithms for each problem and their special cases. Computational study for the algorithms is given in section (5). The conclusion is given in section (6).

2. Notation, basic concepts and mathematical forms

2.1 Notation and basic concepts

The following notation will be used in this paper:

n = number of jobs.
p_i = processing time of job i.
d_i = due date of job i.
C_i = completion time of job i.
E_i = the earliness of the job i.
V_i= the late work penalty for job i.
V_{max} = Max $\{V_i\}$, the maximum late work.
E_{max}= Max $\{E_i\}$, the maximum early work.
f_{max}= Max$\{f_i\}$, the maximum function.
BAB = branch and bound.
LB = lower bound.
UB = upper bound.

In this paper, we shall use the following sequencing rules and concepts:

MST: Jobs are sequenced in non-decreasing order of ($S_i = d_i - P_i$), this rule is well known to minimize E_{max} for $1//E_{max}$ problem [5].

Definition(1):

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 [7].

Definition(2):

A **feasible schedule** σ is **Pareto optimal**, or non-dominated (efficient), with respect to the performance criteria f and g if there is no feasible schedule π such that both $f(\pi) \le f(\sigma)$ and $g(\pi) \le g(\sigma)$, where at least one of the inequalities is strict [11].

Definition(3):

A measure performance is said to be regular if it is a non-decreasing function of job completion times. Example of regular measures are $\sum Ci$,C_{max} ,T_{max} , . . .

Definition(4):

A non-regular performance measure is usually is not a monotone function of the job completion times, an example of such a measure the job earliness [8].

Lawler algorithm (LA):

Step (1): let N ={1,…,n}, Ω=(∅) and M be the set of all jobs with no successors.
Step (2): let j* such that $f_{j*}(\Sigma P_i)$= min{ $f_j(\Sigma P_i)$}, j∈M
Set N = N-{j*} and sequence the job j* in the last position of Ω.
Modify M to represent the new set of the schedule jobs.
Step (3): If N = ∅ stop, otherwise go to step (2).

2.2 The mathematical forms and their algorithms

2.2.1 Hierarchical problems:

We present the mathematical forms and the algorithms for generating solutions when one of two criteria (E_{max} , V_{max}) is more important than the others. These hierarchical problems are also called secondary criteria problems where the secondary criteria refer to the less important criteria. Formulation for multicriteria problems is similar to that for the single criteria problems which require that the optimal value of the primary objective is not violated.

Let us first consider the formulations for bicriteria hierarchical problems say 1// Lex(γ 1 ,γ 2) . There are two parts of the formulations

- Primary objective function (γ 1)
  Subject to:
  Secondary objective function (γ 2)
- Secondary objective function (γ 2)
  Subject to:
  Primary objective function (γ 1)

Hence the algorithm for solving the bicriteria problem needs two steps:
Step (1): We optimize γ 1, followed by
Step (2): The optimization of γ 2, subject to the primary objective value

(γ1). For our problem if E_{max} is more important than V_{max} , then the $1//\ Lex(E_{max}, V_{max})$ problem can be written as:

Min V_{max}
s.t.
$E_{max} = \Delta$, where $\Delta = E_{max}(MST)$.

2.2.2 Simultaneous problems

There are many algorithms that can be used for solving multicriteria scheduling problems, which is to find the efficient solutions or at least approximation to it [3]. The running time for the algorithm often increasing with the increase of the instance size. The purpose of any algorithm process is to find, for each instance a feasible solution called optimal, that minimize the objective function. This usual meaning of the optimum makes no sense in the multicriteria case because it doesn't exist, in most of the cases, a solution optimizing all objectives simultaneously. Hence we search for feasible solutions yielding the best compromise among objectives that constitutes a so called efficient solutions set, these efficient solutions that cannot be improved in one objective without decreasing their performance in at least one of the others. It is clear that this efficient solutions set is difficult to find. Therefore, it could be preferable to have an approximation to that set in a reasonable amount of time.

3. Problem formulation and analysis

The problem of scheduling a set N = {1,…,n} of n jobs on a single machine to minimize multicriteria may be stated as follows. Each job $i \in N$ is to be processed on single machine which can handle only one job at a time, job i has a processing time p_i and due date d_i. All jobs are available for processing at a time zero.

If a schedule $\Omega = (1,\ldots, n)$ is given, then a completion time $C_i = \sum_{j=1}^{i} P_i$ for each job i can be computed and consequently an earliness $E_i = \max\{ d_i - C_i , 0 \}$, $E_{max} = \max\{ E_i \}$ for each i and $E^{w}_{max} = \max\{W_iE_i\}$, where W_i is the important of the job i with respect to other jobs . The late work $V_i(\Omega)$ for the job $i \in N$ which is amount of processing preformed on job i after its due date d_i is easy to compute,

- If $V_i = 0$,then job i is early with $c_i \le d_i$
- If $0 < V_i < p_i$, then job i is partially early
- If $V_i = p_i$,then job i is late with $c_i \ge d_i + p_i$

This means that

$$V_i = \begin{cases} 0 & \text{if } c_i \le d_i ,\ i = 1, \ldots , n \\ c_i - d_i & \text{if } d_i < c_i < d_i + p_i\ i = 1, \ldots , n \\ p_i & \text{if } c_i \ge d_i + p_i ,\ i = 1, \ldots , n \end{cases}$$

Hence $V^{w}_{max} = \max \{ W_i V_i \}$, W_i is the important of job i with respect to other jobs. Our object is to find a schedule that minimizes bicriteria for the following problems:

1. $1//Lex(V_{max},E_{max})$problem (P1) $\in$C1
2. $1//Lex(V^{w}_{max},E_{max})$problem (P2) $\in$C1
3. $1//F(V_{max} , E_{max})$ problem (P3) $\in$ C3
4. $1//F(E^{w}_{max},V_{max})$ problem (P4) $\in$ C3
5. $1//V_{max}+ E_{max}$ problem (P5) $\in$ C2

3.1 1//Lex(V$_{max}$, E$_{max}$) problem (P1)

This problem can be written as:
Min $E_{max} = \max \{ E_i \}$, $\forall$ i
S.t $V_{max} = \Delta$,where $\Delta = V_{max}$ (LA)
Algorithm (1) for problem (P1)

Step(0): Using Lawler algorithm (LA) to find optimal V_{max} and set $\Delta = V_{max}$.

Step(1): Set N={1,…,n}, $t = \Sigma p_i\ \forall i \in N$.

Step(2): Solve $1/ V_i \le \Delta /\ V_{max}$ problem to determine job j to be the job completed at time t,
such that: 1) $V_j \le \Delta$
2) $S_j \ge S_i\ \forall j,i \in N$ and $V_i \le \Delta$.

Step(3): Schedule j in the interval $[t - p_j , t]$

Step(4): Set $N = N - \{j\}$, $t = t - p_j$

Step(5): If $t > 0$, then go to step (2)
Step(6): Stop .

Example (1):
Consider the problem (P1) with the following data:
$P_i = (2, 3, 5, 7)$, $d_i = (11, 7, 18, 9)$ and $i = 1,2,3,4$
Lawler algorithm (LA) gives the sequence (2,4,1,3),with $V_{max}=1$ & $E_{max}=4$
Set $\Delta = 1$, we get the sequence (2,4,1,3) gives $V_{max} = 1$ & $E_{max} = 4$. This sequence is optimal since the optimal sequence (2,4,1,3) with $V_{max} = 1$, $E_{max} = 4$ is obtained by complete enumeration method.

3.2) 1// Lex(V^w_{max},E_{max}) problem (P2)
This problem (P2) can be written as:
Min $E_{max} = \text{Max}\ \{ E_i \}$, $\forall\ i \in N$
S.t $V^w_{max} = \Delta$, where $\Delta = V^w_{max}$ (LA)
Algorithm (2) for problem (P2)

Example (2):
Consider the problem (P2) with the following data:
$P_i = (4, 6, 2, 5)$, $d_i = (20, 9, 4, 7)$, $w_i = (4, 6, 2, 5)$, and $i = 1,2,3,4$
Lawler algorithm (LA) gives the sequence (4,2,3,1)
With $V^w_{max} = 12$ and $E_{max} = 3$
Set $\Delta = 12$,we get the sequence (4,2,3,1) gives $V^w_{max} = 12$ and $E_{max} = 3$ is optimal,

Algorithm (2)
Step(0): Using Lawler algorithm (LA) to find V^w_{max} and set $\Delta = V^w_{max}$.
Step(1): Set $N=\{1,\dots,n\}$, $t = \Sigma\, p_i\ \forall\ i \in N$
Step(2): Solve $1/\ V^w_i \le \Delta\ /\ V^w_{max}$ problem where $V^w_i = W_i V_i$ to determine job j to be the job completed at time t, such that:
1) $W_j V_j \le \Delta$
2) $S_j \ge S_i\ \forall\ j,i \in N$ and $W_iV_i \le \Delta$.
Step(3): Schedule j in the interval $[t - p_j , t]$
Step(4): Set $N = N - \{j\}$, $t = t - p_j$
Step(5): If $t > 0$, then go to step (2)
Step(6): Stop .
since sequence (4,2,3,1) with $V^w_{max} = 12$, $E_{max} = 3$ is also obtained by complete enumeration method.

3.3) 1//F(Vmax,Emax) problem (P3)
Multicriteria scheduling refers to the scheduling problem in which advantages of a particular schedule are evaluated using more than one performance criterion. Several scheduling problem considering the simultaneous minimization of various forms of sum completion time, earliness and tardiness costs have been studied in the literature [(6)], also solves $1//F(f_{max}, g_{max})$ and solves the general problem $1//F(f^1_{max},\dots,f^k_{max})$, K is finite integer number and each one of these functions is assumed to be non-decreasing in the job completion time. Now consider the multicriteria problem $1//F\ (V_{max}, E_{max})$ in which E_{max} is not non_decreasing in job completion time. This problem belongs to C3 and is written as:
Min $\{V_{max} , E_{max}\}$
s.t.
$V_i = \text{Min}\ \{P_i, T_i\}$, $i=1,\dots, n$
$E_i \ge d_i - C_i$, $i=1,\dots, n$
$E_i \ge 0$, $i=1,\dots, n$
$T_i \ge C_i - d_i$, $i=1,\dots, n$
$T_i \ge 0$, $i=1,\dots, n$
The following algorithm (3) is used to solve the problem (P3)

Algorithm(3)
Step(0): Determine the point (V^*_{max},E_{max}) and (V_{max} ,E^*_{max}) by solving $1/\ /V_{max}$ by Lawler algorithm (LA) and $1//E_{max}$ by MST rule. Let SE be the set of efficient (Pareto) solutions, set SE $=\{(V^*_{max} , E_{max}) , (V_{max} ,E^*_{max}) \}$ if each point is not dominated by the other.
Step(1): Set $\Delta = V_{max}$ (MST)

Step(2): Solve $1/ V_i \leq \Delta / V_{max}$ problem by using Lawler algorithm (break tie to schedule the job j last with maximum $S_j = d_j - p_j$; let (V^L_{max} , E^L_{max}) denote the outcome.
Add (V^L_{max} , E^L_{max}) to the set of Pareto optimal points (SE), unless it is dominated by the previously obtained Pareto optimal points.
If SUM is greater than $V^L_{max} + E^L_{max}$ then set SUM=$V^L_{max} + E^L_{max}$.
Let $\Delta = V^L_{max} - 1$ if $\Delta > 0$ repeat step (2), otherwise go to step (3).
Step(3): The Pareto optimal set SE has been obtained and SUM which is the minimum of values for the Pareto points in the set SE.
Step(4): Stop.

Example (3):
Consider the problem (P3) with the following data:
P_i = (10,3,1,5,7), d_i = (16,7,9,8,11) and i = 1,2,3,4 ,5
MST gives the schedule (4,5,2,1,3) with $E^*_{max} = 3$ & $V_{max} = 9$; (V_{max},E^*_{max}) = (9,3).
Lawler algorithm gives the schedule (5,1,4,2,3) with E_{max} =4 & V^*_{max} =5 ; (V^*_{max} , E_{max}) = (5,4). Set SE= {(9,3),(5,4)} & SUM = 9, set Δ = 9 we get the schedule (4,5,2 ,1,3**)** which gives E_{max} = 3 & V_{max} = 9, then the set SE remains the same Let Δ = 9-1=8, we get the schedule (4,2,1,5,3) gives E_{max} =3 & V_{max} =7; (V_{max}, E_{max}) = (7, 3). Then the set SE = {(7, 3), (5, 4)}. Let Δ = 7-1= 6, we get the schedule (4,5,1,2,3) gives E_{max} = 3 & V_{max} = 6; (V_{max}, E_{max}) = (6,3). Then the set SE = {(6,3), (5,4)}. Let Δ = 6-1= 5, we get the schedule (5,1,4,2,3) gives E_{max} = 4 & V_{max} =5; (V_{max}, E_{max}) = (5, 4),then the set SE remains the same. Let Δ = 5-1= 4, There is no $V_j \leq \Delta$, then we stop. The Set of efficient solutions is SE = {(6,3), (5,4)}. & SUM = 9. This set is the set of all efficient which is given by complete enumeration method.

Note that algorithm (3) does not find all the efficient solutions, but it finds most of them as shown in the following example.
Example (4):
Consider the problem (P3) with the following data:
P_i = (7, 14, 3, 1), d_i = (16, 20, 8, 2) and i = 1,2,3,4
The algorithm (3) gives SE={(7,4),(3,8),(5,5)}, but the exact set of efficient solutions which is obtained by complete enumeration method is
SE = {(7, 4), (3, 8), (5,5) , (4,6)},

Note: We can use BAB method to find the set of all efficient solutions.

3.4) 1//F(E^w_{max}, V_{max}) problem(P4)

This problem is denoted by:
Min {E^w_{max} , V_{max}}
St:
$V_i = \min \{ P_i , T_i \}$ i=1,…,n
$W_i E_i \geq W_i (d_i - C_i)$ i=1,…,n
$W_i E_i \geq 0$ i=1,…,n
$T_i \geq C_i - d_i$ i=1,…,n
$T_i \geq 0$ i=1,…,n
The following algorithm (4) is used to solve the problem (P4):

Algorithm(4)
Step(0): Determine the point (E^w_{max} ,V^*_{max}) and (E^{w*}_{max} ,V_{max}) by solving $1/ / V_{max}$ by Lawler algorithm (LA) and $1/ / E^w_{max}$ by WMST rule. Let SE be the set of efficient (Pareto) solutions, set SE ={(E^w_{max} ,V^*_{max}) ,(E^{w*}_{max} ,V_{max})}.
Step(1): Set $\Delta = V_{max}$ (WMST)
Step(2): Solve $1/ V_i \leq \Delta / V_{max}$ problem by using Lawler algorithm (break tie to schedule the job j last with maximum $S_jW_j= (d_j - p_j)W_j$; let ($E^{w(L)}_{max}$, V^L_{max}) denote the outcome.
Add ($E^{w(L)}_{max}$, V^L_{max}) to the set of Pareto optimal Points (SE), unless it is dominated by the previously obtained Pareto optimal points.

Let $\Delta = V^{L}_{max} - 1$ if $\Delta > 0$ repeat step (2), otherwise go to step (3).

Step(3): The Pareto optimal set SE has been obtained with values for the Pareto points.

Step(4): Stop.

Note since the $1//E^{w}_{max}$ problem cannot always solved to optimality by WMST $\{S_iW_i= (d_i - p_i)W_i\}$ rule, hence the algorithm (4) does not gives the set of all efficient solutions.

Example (5):

Consider the problem (P4) with the following data:

$P_i = (7, 3, 2, 7)$, $d_i = (15, 9, 4, 16)$, $W_i = (6, 3, 12, 1)$ and $i = 1,2,3,4$

The WMST gives the sequence (4, 2, 3, 1) with $E^{w}_{max} = 9$ & $V_{max} = 4$, $(E^{w*}_{max}, V_{max}) = (9,4)$ and Lawler algorithm (LA) gives the sequence (2, 1, 4, 3) with $E^{w}_{max} = 30$ & $V^{*}_{max} = 2$, $(E^{w}_{max}, V^{*}_{max}) = (30, 2)$.

Then SE = {(9, 4),(30, 2)}.

Set $\Delta = V_{max}$ (WMST) = 4, we get the sequence (4, 2, 3, 1) gives $E^{w}_{max} = 9$ & $V_{max} = 4$, $(E^{w}_{max}, V_{max}) = (9, 4)$,then the set SE remains the same.

Set $\Delta = 4-1=3$, we get the sequence (4, 2, 1, 3) gives $E^{w}_{max} = 9$ & $V_{max} = 2$, $(E^{w}_{max}, V_{max}) = (9, 2)$. Then SE = {(9, 2)}.

Set $\Delta = 1$, There is no $V_j \le \Delta$, then we stop. The set of efficient solutions is SE = {(9, 2)}.

4. 1\\(Vmax+Emax) problem(p5):

The aim for problem (P5) is to find a processing order σ of the jobs on a single machine to minimize the sum of maximum earliness and the maximum late work (i.e. to minimize $V_{max}(\sigma)+E_{max}(\sigma)$, $\sigma\in S$ where S is the set of all feasible solutions). It is clear that the problem (P5) is a special case of the problem (P3).

In this section we decompose the $1//V_{max}+ E_{max}$ problem into two subproblems with a simpler structure. For this problem let:

$M=\min_{\sigma\in S}\{V_{max}(\sigma)+ E_{max}(\sigma)\}$.

The problem (P5) can be decomposed into two subproblems (SP_1) and (SP_2).

$M_1 = \min\{\max\{V_{\sigma(i)}\}\}$.

s.t

$$V_{\sigma(i)} = \left\{\begin{array}{lll} 0 & \text{if} \quad C_{\sigma(i)} \le d_{\sigma(i)}, & i=1,\dots,n \\ C_{\sigma(i)} - d_{\sigma(i)} & \text{if} \quad d_{\sigma(i)} < C_{\sigma(i)} < d_i + p_{\sigma(i)}, & i=1,\dots,n \\ p_{\sigma(i)} & \text{if} \quad d_{\sigma(i)} + p_{\sigma(i)} \le C_{\sigma(i)}, & i=1,\dots,n \end{array}\right\}$$

… (SP_1)

$V_{\sigma(i)}=\min\{C_{\sigma(i)}-d_{\sigma(i)}, P_{\sigma(i)}\}$, $i=1,\dots,n$

$M_2 = \min\{\max\{E_{\sigma(i)}\}\}$.

s.t.

$E_{\sigma(i)}=d_{\sigma(i)}-C_{\sigma(i)}$, $i=1,\dots,n$ … (SP_2)

$E_{\sigma(i)} \ge 0$, $i = 1,\dots,n$

4.1 Derivation of Lower Bound (LB) for Problem (P5)

The lower bound (LB) is based on decomposing problem (P5) into two subproblems (SP_1) and (SP_2). Then calculate M_1 to be the minimum value for (SP_1) and M_2 to be the minimum value for (SP_2) then applying the following theorem:

Theorem (1):

$M_1+M_2\le M$ where M_1, M_2, and M are the minimum objective function values of (SP_1), (SP_2), and (P5) respectively [1].

To get a lower bound LB for the problem (P5):

For the subproblem (SP_1) we compute M_1 to be a lower bound by sequencing the jobs using Lawler's algorithm (LA) to find the minimum maximum late work V_{max} .

For the subproblem (SP_2) we compute M_2 to be a lower bound by sequencing the jobs by MST order (i.e sequencing the jobs in non- decreasing order of $S_i = d_i - P_i$) to find the minimum maximum early job E_{max}, then applying theorem (1) to obtain:

$LB = M_1 + M_2$

4.2 Heuristic Method to Calculate Upper Bound (UB) for the problem (P5)

A simple heuristic is obtained by sequencing the jobs using Lawler's algorithm (LA) to find V^*_{max} and E_{max}, then

$UB_1 = V^*_{max}$ (LA) + E_{max} (LA).

UB_2 is obtained by ordering the jobs in MST order, that is, sequencing the jobs i,(i=1,…,n) in non-decreasing order of $S_i = d_i - P_i$ to find E^*_{max} and V_{max} ,then

$UB_2 = V_{max}$ (MST)+ E^*_{max} (MST).

Then UB = min {UB_1, UB_2}.

4.3 Branch and bound (BAB) method

Our BAB method based on forward sequencing branching rule for which nodes at level k of the search tree corresponding to initial partial sequence in which jobs are sequenced in first k positions. The LB at any node is the cost of scheduling jobs (this cost depends on the objective function) and the cost of unsequenced jobs (this cost depends on derived lower bound (LB)). At any level of the BAB method, if a node has LB $\geq$ UB, then this node is dominated.

If the branching ends at a complete sequence of jobs then this sequence is evaluated, and if its value is less than the current (UB), this (UB) is reset to take that value. The procedure is then repeated until all nodes have been consider by using back tracking procedure. Backtracking procedure is the movement from the lowest level to the upper level in the BAB method. The (UB) at the end of this procedure is the optimum for our scheduling problem (P5). Hence by using BAB method we get at least one optimal solution. It is clear that the BAB method is improved by using efficient (LB), good (UB) and dominance rule. If it can be shown that an optimal solution can be always generated without branching from a particular node of the search tree, then that node is dominated and can be eliminated. Dominance rules usually specify whether a node can be eliminated before its (LB) is calculated. Clearly, dominance rules are particularly useful when a node can be eliminated which has a (LB) that is less than the optimal solution.

<u>Example (6):</u>

Consider the problem (P5) with the following data:

$P_i = (2, 5, 7, 4)$, $d_i = (6, 21, 10, 9)$ and i = 1,2,3,4

Lawler's algorithm gives a schedule (4, 3, 1, 2) where $V^*_{max} = 2$ and $E_{max} = 5$, then $UB_1 = V^*_{max}$ (LA) + E_{max} (LA) = 2+5 = 7

And MST rule gives a schedule (3, 1, 4, 2), where V_{max} (MST) =4 &

E^*_{max} (MST) =3, then $UB_2 = V_{max}$ (MST) + E^*_{max} (MST) = 4+3 = 7

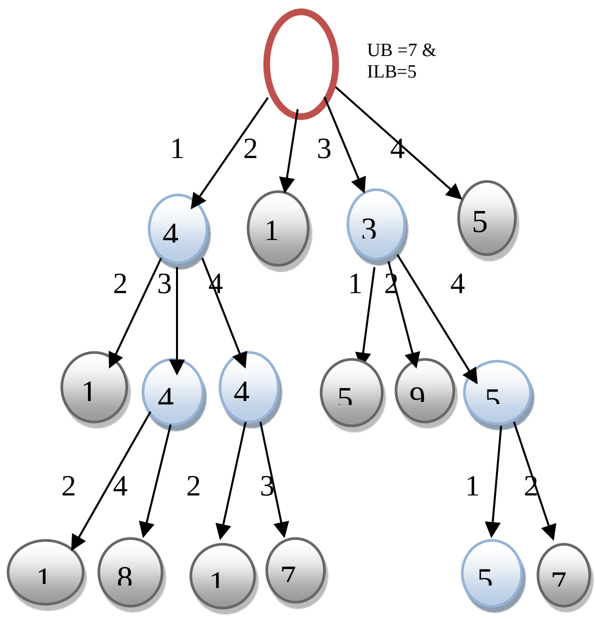
Hence the minimum upper bound is

UB= min {UB_1, UB_2} = 7

$LB_1 = V^*_{max}$ (LA) = 2 & $LB_2 = E^*_{max}$ (MST) = 3

$ILB = LB_1 + LB_2 = 2+3 = 5$

We now give the BAB tree algorithm to find optimal solution for (P5)



Note:
If MST rule gives (V_{max},E^*_{max}) and Lawler algorithm (LA) gives (V^*_{max},E_{max}) and $E_{max} = E^*_{max}$ and $V_{max}=V^*_{max}$ then there exists an optimal solution with value (V^*_{max} ,E^*_{max}).

The first result for dominance rule is given next.

Lemma (1):
If there exist a job i with max (S_i) and min ($V_i(t)$), $t=\sum_{i\in N} P_i$ then there exists on optimal schedule in which job i is sequenced last for $1//V_{max}+E_{max}$ problem.

Proof:
Suppose we sequence the jobs in MST rule (i.e non_decreasing of S_i) to minimize E_{max} **,** then job i is sequence last with minimum E_i ,since it has maximum S_i .

Now if we use Lawler's algorithm (LA) to minimize V_{max} , then we find job i is sequence last because it has a minimum late work V_i .Hence there exist an optimal schedule for $1//V_{max}+E_{max}$ problem in which job i is sequenced last. ■

Note: Problem reduction for problem (P5)

In the case of our problem $1//V_{max}+E_{max}$, there exists n! possible orders of the jobs. We try to reduce the size of the problem by finding the job which succeeds all other jobs in an optimal schedule. Such a job is removed and hence the number of possibilities is reduced if lemma (1) above is satisfied. When no further progress can be made with reduction condition we use BAB method to find optimal solution.

Note that sometimes special cases for NP-hard scheduling problem which is satisfying some conditions make the problem easily solvable.

Some Special case for $1//V_{max}+E_{max}$ problem(P5)

Case (1)

The problem (P5) is solved for special case, if $d_i = d\ \forall\ i$ then there exists an optimal schedule σ obtained by Lawler's algorithm (LA) in which $d < C_1 < \cdots < C_n$ where C_1, C_n are minimum completion time and maximum completion time respectively:

Proof: We have three positions for $d_i = d$ to be considered:

(1) $C_1 < d < C_n$ then there exists an optimal schedule with value $V_{max}+E_{max} = V^*_{max} + d - C_1$ where V^*_{max} obtained by using Lawler's algorithm.

(2) $d < C_1 < \ldots < C_n$ then there exists an optimal schedule with value $V_{max}+E_{max} = V^*_{max} + 0 < V^*_{max} + d - C_1$ where V^*_{max} obtained by using Lawler's algorithm.

(3) $C_1 < \ldots < C_n < d$ then there exists an optimal schedule with value $V_{max}+E_{max} = d - C_1 > V^*_{max}$

Hence Lawler's algorithm gives minimum V^*_{max} for $V_{max}+E_{max}$.

5. Experimental Results

Experimental Results of algorithm (3) and BAB for the problems (P3) and (P5) respectively. Table (1) shows the results of applying algorithm (3) in order to get a set of efficient solutions and minimum sum of V_{max} and E_{max} for the problem (P3), on samples of different jobs with 5 experiments for each. The results of efficient solutions compared with the results obtained from complete enumeration method (CEM), which generates all solutions for $n \leq 7$.

Table (1) the set of efficient solutions obtained by algorithm (3) and complete enumeration method

n	EX	Efficient solution (E_{max} , V_{max})		
		Algorithm (3)	CEM	SUM
3	1	(2,2) , (1,3)	(2,2) , (1,3)	4
	2	(2,4)	(2,4)	6
	3	(3,3) , (2,6)	(3,3) , (2,6)	6
	4	(2,5)	(2,5)	7
4	1	(3,6) , (2,8)	(3,6) , (2,8)	9

	2	(1,3)	(1,3)	4
	3	(3,4) , (2,6)	(3,4) , (2,6)	7
	4	(2,4)	(2,4)	6
	5	(6,5)	(6,5)	11
5	2	(0,4)	(0,4)	4
	3	(3,4)	(3,4)	7
	4	(3,9) , (0,10)	(3,9) , (0,10)	10
	5	(0,7)	(0,7)	7
	1	(2,3) , (0 4)	(2,3) , (0 4)	4
6	2	(1,5)	(1,5)	6
	3	(3,1)	(3,1)	4
	4	(8,0)	(8,0)	8
	5	(2,6) , (1,8)	(2,6) , (1,8)	8
	2	(0,4)	(0,4)	4
7	1	(3,7) , (0,8)	(3,7) , (0,8)	8
	2	(3,8)	(3,8)	11
	3	(0,5)	(0,5)	5
	4	(10,0)	(10,0)	10
	5	(3,7) , (2,8)	(3,7) , (2,8)	10
8	1	(3,7) , (1,8)		9
	2	(3,5) , (2,6)		8
	3	(3,6) , (2,7)		9
	4	(7,4) , (3,8)		11
	5	(3,5)		8
9	1	(2,8) , (1,9)		10
	2	(3,7) , (1,8)		9
	3	(0,7)		7
	4	(15,0)		15
	5	(13,1)		14
10	1	(13,5)		18
	2	(2,5) , (1,8)		7
	3	(0,9)		9
	4	(17,2)		19
	5	(2,9) , (3,7)		10
11	1	(3,7)		10
	2	(6,8)		14
	3	(0,6)		6
	4	(0,8)		8
	5	(1,8) , (0,9) , (3,7)		9
12	1	(3,7)		10
	2	(2,4)		6
	3	(4,6)		10
	4	(3,6) , (2,8)		9
	5	(2,6) , (0,8) , (3,5)		8
13	1	(14,5)		19
	2	(3,6)		9
	3	(1,6) , (0,7)		7
	4	(32,0)		32
	5	(7,8) , (1,9)		10
14	1	(0,9)		9
	2	(0,10)		10
	3	(3,9)		12
	4	(17,4)		21
	5	(8,5)		13
15	1	(18,3)		21
	2	(14,4)		18
	3	(3,9)		12
	4	(16,2) , (14,4)		18
	5	(8,1)		9
16	1	(3,8) , (1,9)		10
	2	(4,6) , (1,7)		8
	3	(3,8) , (2,9)		11
	4	(0,10)		10
	5	(3,7) , (0,8)		8
17	1	(21,5) , (0,10)		10
	2	(7,8) , (0,9)		9
	3	(3,7)		10

	4	(2,10)		12
	5	(8,6) , (2,7)		9
19	1	(23,1)		24
	2	(9,4)		13
	3	(25,0)		25
	4	(3,10)		13
	5	(3,9) , (0,10)		10

Notes:

1) The results for n>7 for complete enumeration cannot be obtained.

2) The minimum SUM = $E_{max}+V_{max}$ for n≤7 for the algorithm(3) is the same as in complete enumeration method.

Table (2) shows the results of applying BAB method on different values of n of problem (P5) to get minimum sum of V_{max} and E_{max} .

Table (2): The optimal solution for the 1// V_{max}+ E_{max}

n	EX	BAB	n	EX	BAB	n	EX	BAB
3	1	4	4	1	9	5	1	6
	2	6		2	4		2	4
	3	6		3	7		3	7
	4	7		4	6		4	10
	5	4		5	11		5	7
6	1	4	7	1	8	8	1	9
	2	6		2	11		2	7
	3	4		3	5		3	9
	4	8		4	10		4	11
	5	8		5	10		5	8
9	1	10	10	1	18	11	1	10
	2	9		2	7		2	14
	3	7		3	9		3	6
	4	15		4	19		4	8
	5	14		5	10		5	9
12	1	10	13	1	19	14	1	9
	2	6		2	9		2	10
	3	10		3	7		3	12
	4	9		4	32		4	21
	5	8		5	10		5	13
15	1	21	16	1	10	17	1	9
	2	18		2	8		2	9
	3	12		3	11		3	10
	4	18		4	10		4	12
	5	9		5	8		5	9
19	1	24						
	2	13						
	3	25						
	4	13						
	5	10						

Note: In table (2) BAB means the optimal sum of V_{max}+ E_{max} for (P5)

The best sum of V_{max} and E_{max} for problem (P3) obtained from algorithm (3) results are compared with the BAB method results for different number of jobs, five experiment for each number of jobs for problem (P5) are given in table (3).

Table (3): Comparison of results of minimum SUM of algorithm (3) with the results of BAB method .

n	EX	BAB	n	EX	BAB	n	EX	BAB
3	1	4	4	1	9	5	1	6
	2	6		2	4		2	4
	3	6		3	7		3	7
	4	7		4	6		4	10
	5	4		5	11		5	7
6	1	4	7	1	8	8	1	9
	2	6		2	11		2	7
	3	4		3	5		3	9
	4	8		4	10		4	11
	5	8		5	10		5	8
9	1	10	10	1	18	11	1	10
	2	9		2	7		2	14
	3	7		3	9		3	6
	4	15		4	19		4	8
	5	14		5	10		5	9
12	1	10	13	1	19	14	1	9
	2	6		2	9		2	10
	3	10		3	7		3	12
	4	9		4	32		4	21
	5	8		5	10		5	13
15	1	21	16	1	10	17	1	9
	2	18		2	8		2	9
	3	12		3	11		3	10
	4	18		4	10		4	12
	5	9		5	8		5	9
19	1	24						
	2	13						
	3	25						
	4	13						
	5	10						

6. Conclusions

From our Computational results we conclude that:

1) The number of efficient solutions (points) of algorithm (3) is less than the number of jobs n.
2) Algorithm (3) can find most of efficient points and this clear from the results of Table (3) from the 80 test problems for n = 3,4,…,19 only for n = 8, the experiment (2) gives SUM = 8 and the exact SUM = 7 which is obtained by BAB method and n=17,the experiment (1) gives SUM =10 and the exact SUM =9 which is obtained by BAB method.
3) The algorithm (3) can be used for solving problems of the form 1// F(V_{max}, F_{max}).
4) The problem (P5) can be solved by using BAB method to get optimal solution for $n \leq 19$. Since the problem (P5) is a special case of problem (P3), hence the algorithm (3) can be used to find near optimal solutions without using BAB method and in reasonable time for large n.

References

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

(2) Chang P., Su L., (2001). Scheduling n jobs on one machine to minimize the maximum lateness with a minimum number of tardy jobs. Computer & industrial engineering, 40, 49-60.

(3) Hoogeveen, J.A ., (2005). Invited Review Multicriteria scheduling. European Journal of Operational Research 167 ,592–623(2005).

(4) Hoogeveen, J.A., van de Velde, S.L., (1995). Minimizing total completion time and maximum cost simultaneously is solvable in

polynomial time. Operations Research Letters 17, 205–208.
(5) Hoogeveen, J.A., (1992). Single-machine bi-criteria scheduling. PhD Dissertation, Center for mathematics and Computer science, Amsterdam. The Netherlands.
(6) Hoogeveen, J.A., (1996). "Single machine scheduling to minimize a function of two or three maximum cost criteria". Journal of Algorithms 21, 415– 433.
(7) Jouni L., (2000). Multi-objective Nonlinear Pareto-Optimization. Lappeenranta University of Technology.
(8) Lawler E.L., (1973). Optimal sequencing of a single machine subject to precedence constraint. Management Science 19/5, 544-546.
(9) Moslehi G., Moghaddam R., Vasei M., and Azaron A., (2005). Optimal scheduling for a single machine to minimize the sum of maximum earliness and tardiness considering idle insert. Applied Mathematics and Computation 167 , 1430–1450.
(10) Nagar A., Jorge H., and Sunderesh H., (1995). Multiple and bi-criteria Scheduling: A literature survey. European Journal of Operational Research North-Holland, 81, 88-104.
(11) Van Wassenhove L.N., and Gelders F., (1980). Solving a bicriterion scheduling problem. European Journal of Operational Research 4/1, 42-48.

حل مسائل الجدولة ذات دوال هدف متعددة

طارق صالح عبد الرزاق و كرار فتاح عبد الرزاق

قسم الرياضيات–كلية العلوم – الجامعة المستنصرية

الخلاصة

في هذه الأطروحة اقترحنا خوارزميات لمسألة متعددة المعاييرعلى ماكنة واحده لتصغير معيارين : تكلفة الأعمال المتأخره التعاظمية ودالة التبكير التعاظمية. نقترح عدة خوارزميات تعتمد على أنواع من الاهداف تعمل على الوجه الأمثل. تتم مقارنة الحلول من الإجراءات المقترحة مع الحلول المثلى والحلول باريتو الأمثل لحجم صغير على سبيل المثال،هذه الخوارزميات تتعامل مع مسألة التصغير الهرمي وكذلك مسالة الأهداف المتساويه من حيث الاهمية في وقت واحد مع وبدون وزن. تظهر النتائج الحسابية فائدة هذه الإجراءات.