SCITECH
RESEARCH ORGANISATION

Volume 7, Issue 1
Published online: March 18, 2016|

Journal of Progressive Research in Mathematics
www.scitecresearch.com/journals

Using Branch and Bound Method to Minimize Bi-Criteria

AL_ Zuwaini Mohammad Kadhim, Asmaa Ali Zeyad

Department of Mathematics, College of Computer Science and Mathematics,Thi-QarUniversity, Thi-Qar,Iraq.

Mathematic Dept, College of Education for Pure Science, Thi-Qar University, Thi-Qar, Iraq.

ABSTRACT

This paper presents a branch and bound algorithm for sequencing a set of n independent jobs on a single machine to minimize sum of total late work and the number of tardy jobs, the type of the problem is NP-hard. Lower bounds were proposed and heuristic method to get an upper bound. Some special cases were proved and some dominancerules were proposed and proved, the problem solved with up to 40 jobs.

Keywords: Late work; Tardy jobs; Scheduling.

1. INTRODUCTION

We had study the scheduling independent njobs on a single machine to minimize the bi criteria problem. Our objective is to find a schedule that minimize sum of total late work and the number of tardy jobs when all jobs are available at time zero, by using (BAB) method. This problem is denoted by $1//\sum_{j=1}^{n}(V_j + U_j)$ However, this problem consist of two sub problems.

The first sub problem is$1//\sum_{j=1}^{n} V_j$ is NP-hard in non-preemptive[1]. The parameter V_j was initially introduce by Blazewicz (1984)as information loss, Blazewicz considered the problem of sensors collecting data for parallel processors. If the control system receives the data after a set deadline, it results in a complete data loss [2]. In The preemptive case;$1/pmtn/\sum_{j=1}^{n} V_j$ of the problem issolved by O (n logn) algorithm [1].Some researchers studies the $\sum V_j$ problem on single machine (Hariri, Potts, and Van Wassenhove,1995[3]; Kethley and Alidaee, 2002[4]), multiple machine scheduling (Błażewicz, Pesch, Sterna, and Werner,2005a[5], 2005b[6], 2007[7],2008[8]; Lin, Lin, & Lee, 2006[9]). Al-Zuwaini (2006) [10] used branch and bound technique with a suitable lower bound and proved some dominance rules for $1/r_j/V_j$ problem. Other researchers studied this criteria (V_j) with other criteria to consider multi scheduling for example$(\sum V_j + \sum C_j)$[11]. Al-Nuaimi (2014)[12] used efficient branch and bound technique with a suitable lower bound and proved some dominance rules for $1//F(\sum_{j=1}^{n} V_j, V_{max})$.

In the second sub problem; we can note that minimizing the number of tardy jobs on a single machine ($1\text{ha}\sum_{j=1}^{n} U_j$) is solved in O(n log(n)) time (Moore, 1968)[13]. This algorithm repeatedly adds jobs in EDD order to the end of partial schedule of on time jobs. If the addition of job j results in this job being complete after time d_j, then a job in the partial schedule with the largest processing time is removed and declared late [14].

2. Formulation of problem

A set of n independent jobs $\{J_1, J_2, \cdots, J_n\}$ be schedule on a single machine only one job$j$ can be processed at a time. All jobs are available for processing at time zero. The goal is to find a processing order of the jobs that minimize the total late works and number of tardy jobs, which is denoted by$1 // \sum_{j=1}^{n}(V_j + U_j)$. In this problem precedence relation between jobs is not supposed and preemption is not allowed. Each job j has positive integer processing time (p_j) and due date (d_j), the completion time($C_j = C_{j-1} + p_j,\ and\ C_0 = 0$) if j completed after due date($C_j > d_j\ then\ U_j = 1$) job is said late otherwise; ($C_j \leq d_j$)$then\ U_j = 0$. Our objective is finding a schedule to minimize the sum of number of tardy job sand total late works which are given by:

$V_j = \min\{T_j, p_j\}$

And

$$U_j = \begin{cases} 1 & if\ C_j > d_j \\ 0 & if\ C_j \leq d_j \end{cases}$$

Since $1 // \sum_{j=1}^{n} V_j$ is NP-hard [1] then the problem under investigation is known to be NP-hard. The objective is to find the schedule $\sigma = (\sigma(1), \dots, \sigma(n))$ of the jobs that minimize the total cost W, this denoted by (P) can be state as follows:

$$\left.\begin{aligned} W &= min_{\sigma\in\delta}\left\{\sum_{j=1}^{n}(V_{\sigma(j)} + U_{\sigma(j)})\right\} \\ s.t& \\ C_{\sigma(j)} &\geq p_{\sigma(j)} && j = 1, \dots, n \\ V_{\sigma(j)} &\leq C_{\sigma(j)} - d_{\sigma(j)} && j = 1, \dots, n \\ V_{\sigma(j)} &\leq p_{\sigma(j)} && j = 1, \dots, n \\ U_{\sigma(j)} &\geq 0 && j = 1, \dots, n \\ U_{\sigma(j)} &\leq 1 && j = 1, \dots, n \\ p_{\sigma(j)} &> 0, d_{\sigma(j)} > 0,\ V_{\sigma(j)} \geq 0 \end{aligned}\right] \dots (P)$$

Where δ set of all feasible solution.Where $\sigma(j)$ denotes the position of job j in the ordering σ.

3. Decomposition of Problem (p)

In this section we will breakdown the problem (P) into two sub problems $(SP_1)$and (SP_2)which are simple structure of original problem as follow:

$$\left.\begin{array}{ll} W_1 = min_{\sigma\in\delta}\{\sum_{j=1}^{n} V_{\sigma(j)}\} & \\ s.t & \\ C_{\sigma(j)} \geq p_{\sigma(j)} & j = 1,\dots,n \\ V_{\sigma(j)} \leq C_{\sigma(j)} - d_{\sigma(j)} & j = 1,\dots,n \\ V_{\sigma(j)} \leq p_{\sigma(j)} & j = 1,\dots,n \\ V_{\sigma(j)} \geq 0,\ p_{\sigma(j)} > 0,\ d_{\sigma(j)} > 0 & \end{array}\right] \dots (SP_1)$$

And

$$\left.\begin{array}{ll} W_2 = min_{\sigma\in\delta}\{\sum_{j=1}^{n} U_{\sigma(j)}\} & \\ s.t & \\ C_{\sigma(j)} \geq p_{\sigma(j)} & j = 1,\dots,n \\ U_{\sigma(j)} \geq 0 & j = 1,\dots,n \\ U_{\sigma(j)} \leq 1 & j = 1,\dots,n \\ p_{\sigma(j)} > 0,\ d_{\sigma(j)} > 0 & \\ , & \end{array}\right] \dots (SP_2)$$

Theorem (3.1)[14]: $W_1 + W_2 \leq W$ whereW_1 , W_2, and W are the minimum objective function values of SP_1, SP_2 and P respectively .

4. Special case

A machine scheduling problem of type NP-hard is not simply solvable and it is more difficult when the objective function is multi objective.A special case for scheduling problem means finding an optimal schedule directly without using mathematical programming techniques(such as: DP, BAB and complete enumeration method). If the above-mentioned case exists, it's depends on satisfying some conditions in order to make the problem easily solvable. These conditions depend on the objective function as well as the data of jobs [15]. In this section some special cases of problem(P) are givenand proved.

Case (1): If there is a schedule π satisfy $P_j = P \text{ and } d_j > jP \quad \forall\, j \in \pi$, then the schedule π gives optimal solution for $1 / P_j = P / \sum(V_j + U_j)$ problem

Proof: Form the condition of processing time, $C_j = jP$ and $d_j > jp$ then $C_j < d_j$ this mean that job j is early. Hence;$\sum(V_j + U_j) = 0$ and π is optimal.■

Case(2): For the problem(P), if there is a schedule π satisfy$c_j \leq d_j \ \forall\, j\ (j = 1,\dots,n)$then$\pi$ is optimal and $\sum(U_j + V_j) = 0$

Proof: It is clear from case (1).■

Case(3): If (SPT) rulesatisfy $T_j < P_j \quad \forall j$ and $\sum U_j\,(SPT) = \sum U_j\ (MA)$ then (SPT)rule, given an optimal solution for the problem $1/ / \sum(V_j + U_j)$.

Proof: AL-Zuwaini (2006)[10], proved that(SPT) is optimal solution for, $1/T_j < P_j / \sum V_j$problem, but $\sum U_j\,(EDD) = \sum U_j\ (MA)$. Then (SPT)rule gives optimal solution for the problem $1\ //\ \sum(V_j + U_j)$.■

Case(4): If $\sum V_j\ (EDD) = T_{max}\,(EDD)$ and $\sum U_j\,(EDD) = \sum U_j\ (MA)$ then EDD rule gives optimal solution for the problem $1//\sum(U_j + V_j)$

Proof: Since $T_{max}\,(EDD)$ is a lower bound for total late works $(\sum V_j)$[1].Then (EDD) rule optimal for $\sum V_j$. But$1//\sum U_j$ minimized by (MA)[16]. So EDD rule optimal for $(1//\ \sum_{j=1}^{n}(U_j + V_j)$ problem.■

Case(5): If $p_j = p \ and \sum V_j(EDD) = \sum V_j(MA)$ then MA algorithm gives optimal solution for $1 \,//\, \sum_{j=1}^{n}(V_j + U_j)$ problem.

Proof: Since $p_j = p$ then (EDD) rule gives optimal $1 \,//\, \sum_{j=1}^{n} V_j$ [17]. And from $\sum_{j=1}^{n} V_j\ (MA) = \sum_{j=1}^{n} V_j\,(EDD)$, we get (MA) gives optimal for $1 \,//\, \sum(V_j + U_j)$ problem.■

Case(6): If $d_j = d\ \forall j (j = 1, \dots, n)$ then (SPT) gives optimal solution for the problem $1/d_j = d/\sum_{j=1}^{n}(U_j + V_j)$

Proof: Since $d_j = d$.Then for any ordering of the jobs the total late work equal to $\max\{\sum_{j=1}^{n} p_j - d\ ,0\}$. Thus any jobs sequence specifies optimal schedule [17]. But (SPT)rule gives optimal solution for problem$1/d_j = d \,/\, \sum_{j=1}^{n} U_j$. Then (SPT) gives optimal solution for $1/d_j = d \,/\, \sum_{j=1}^{n}(U_j + V_j)$.■

Case(7): If $p_j = p \forall j$ and EDD rule satisfy $\sum U_j(EDD) = \sum U_j(MA)$ then EDD gives optimal solution for $1 \,//\, \sum_{j=1}^{n}(U_j + V_j)$.

Proof: From the condition we get EDD is optimal for $1 \,//\, \sum_{j=1}^{n} U_j$. But Potts and Van [17] prove that EDD gives an optimal for $1 \,//\, \sum_{j=1}^{n} V_j$ provided that $p_j = p\ \forall j$. So (EDD) gives optimal for $1 \,//\, \sum_{j=1}^{n}(U_j + V_j)$ problem.■

Case(8): If $p_j = p$ and $d_j = d$then the problem$1/p_j = p\,, d_j = d \,/\, \sum_{j=1}^{n}(U_j + V_j)$ is independent on a schedule

Proof: Since $C_j = jp$ and $V_j = min\left\{max\left(C_j - \mathrm{d}, 0\right), \mathrm{p}\right\}$in any order. Then the problem is dependent on $1/p_j = p\,, d_j = d/\sum_{j=1}^{n} U_j$ problem and any jobs sequence specifies optimal schedule for $1 \,//\, \sum U_j$ provided that common due date and constant processing time.■

5. Branch and Bound (BAB) Method

Our branch and bound algorithm used forward sequencing branching rule for which nodes at level (L) of the search tree correspond to initial partial sequenced in the first (L) position.

5.1 Upper bound (UB)

We suggested a heuristic method which is applied at the root node of the branch tree to find an upper bound UB on theminimize value of problem (P)

Heuristic (UB)

Step(1): ordered the job according to (EDD) rule to obtain sequence $\pi = (1\,,2\,, \dots, n)$

Step(2): find $T_{max}\,(\pi)$

Step(3): if $T_{max}\,(\pi) = 0$ then $UB = \sum_{j=1}^{n}(V_j + U_j) = 0$ and (π) gives optimal solution. Go to step(7)

$$else$$

$T_{max}\,(\pi) > 0$, choose job $j,\ j \in \pi$ such that

$C_{j-1} < T_{\max}\,(\pi) \leq C_j$Reordered the sequence π such that $\sigma = (j+1, \dots, n, 1, \dots, j-1)$

Step(5): if $C_j > T_{max}\,(\pi)$ then job j in first position of sequence σ.

$else$

Job j in last position of sequence σ.

Step(6):compute UB=$\sum_{j\in\sigma}(V_{\sigma(j)} + U_{\sigma(j)})$

Step(7): stop

Example: find UB to four jobs by the heuristic

Job	**1**	**2**	**3**	**4**
p_j	4	3	6	7
d_j	10	15	13	11

Solution:

$\pi = (1,4,3,2)$, $T_{max}(\pi) = 5 > 0$. $C_1 < T_{max}(\pi) \leq C_4$

Since $T_{max}(\pi) < C_4$ then $\sigma = (4,3,2,1)$, $\sum_{j\in\sigma}(V_j + U_j) = 7$

6. Lower bound

The lower bound for the problem (P) is based on decomposition (P) of two sub problems (SP_1) and (SP_2). Moreover, calculated W_1 and $W_2$to be the lower bounds for (SP_1) and (SP_2) respectively as in section (3), and applying theorem (3.1) to get a lower bound (LB) for problem(P).

For sub problem (SP_1), LB_1=W_1=$T_{max}(EDD)$ is lower bound [1]. And for problem (SP_2), $W_2 = LB_2 = \sum_{j\in MA} U_j$. Hence $LB = LB_1 + LB_2$. Is initial lower bound which it used in root node in search tree.

7. Dominance rule

Because of branching scheme, the size of the search tree is directly linked to the length of the current sequence (which represents the number of nodes). Hence, a preprocessing step is performed in order to remove as many positions as possible. Reducing the current sequence is done by using several dominance rules. Dominance rules usually specify whether a node can be eliminated before its lower bound was calculated. Clearly, dominance rules are particularly useful when a node can be eliminated which has a lower bound that is less than the optimum solution[15] .Some of dominance rules are valid for minimization of the sum of total late works and number of tardy jobs . Below two dominance rules are stated in order to decrease the number of nodes in search tree as well as decreasing the time. We can apply theorem (7.1) when jobs $j\ and\ i$ cannot be early.

***Theorem (7.1)*:** If S_k partial sequence which its job are schedule $k \subset N$ for $i, j \in \bar{k} = N - k$ if $d_i \leq d_j$. Then job $j \prec i$ in optimal solution of problem (P)
Proof:

Let (S_k, i, j) be schedule which obtain by interchange job i and j in (S_k, j, i). then all job other than i and j have same completion time in (S_k, i, j) as in (S_k, j, i).

In (S_k, j, i):

$$V_j = \min\{t + p_j - d_j, p_j\}$$

$$V_i = \min\{t + p_i + p_j - d_i, p_i\}$$

And

$$U_j = \begin{cases} 1 & if\ t + p_j \geq d_j \\ 0 & o.w \end{cases}$$

$$U_i = \begin{cases} 1 & if\ t + p_i + p_j \geq d_i \\ 0 & o.w \end{cases}$$

In (S_k, i, j):

$V_i = \min\{t + p_i - d_i, p_i\}$

$V_j = \min\{t + p_i + p_j - d_j, p_j\}$

And

$U_i = \begin{cases} 1 & if\ t + p_i \geq d_i \\ 0 & o.w \end{cases}$

$U_j = \begin{cases} 1 & if\ t + p_i + p_j \geq d_j \\ 0 & o.w \end{cases}$

There are some case for d_i, d_j with respect to t:

(1) If $d_i \leq d_j \leq t$

Then job $i\ and\ j$ is late in $(S_k, j, i)$and (S_k, i, j)

(i.e.) $\sum_{i \in S_{k,j,i}}(V_i + U_i) = \sum_{i \in S_{k,i,j}}(V_i + U_i) = p_i + p_j + 2$

(2) If $d_i \leq t \leq d_j$

In (S_k, j, i)

$V_j + U_j = t + p_j - d_j + 1$ and $V_i + U_i = p_i + 1$

In (S_k, i, j)

$V_i + U_i = p_i + 1$, $V_j + U_j = \begin{cases} t + p_j + p_i - d_j + 1 \\ p_j + 1 \end{cases}$

Then $\sum_{i \in S_{k,j,i}}(V_i + U_i) \leq \sum_{i \in S_{k,i,j}}(V_i + U_i)$

(3) If $t < d_i \leq d_j$

In (S_k, j, i)

$V_j + U_j = t + p_j - d_j + 1$ and $V_i + U_i = p_i + 1$

In (S_k, i, j)

$V_i + U_i = t + p_i - d_i + 1$, $V_j + U_j = \begin{cases} t + p_j + p_i - d_j + 1 \\ p_j + 1 \end{cases}$

Then $\sum_{i \in S_{k,j,i}}(V_i + U_i) \leq \sum_{i \in S_{k,i,j}}(V_i + U_i)$. So j$\prec i$ in optimal solution for problem (P).

***Theorem (7.2)*:** Suppose that i and j unscheduled jobs with $d_i = d_j\ and\ p_j \leq p_i$. Then job j$\prec i$ in optimal solution for problem (P).

Proof: By case (6).

8. Computational Experience

An intensive work of numerical experimentations has been performed. We present in how instances (test problems) can be randomly generated. The data was generated in this paper in the same way as in[18]. That generates as following:

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

- The due date d_i is uniformly distributed in the interval [P(1-TF-RDD/2),P(1-TF+RDD/2)];where P= $\sum P_i$ 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), for each of the values of parameters producing 10 problems for each values of n.

The BAB algorithm was tested by coding it in MATLAB 7.10.0 (R2010a) and implemented on Intel (R) Core (TM) i7-4500M CPU @ 1.80 GHZ, with RAM 2.40 GB personal computer.

In table (1.1), shows the results for problem (P) obtained by (BAB) algorithm. We list 10 test problems for each value of n, where n ϵ {5, 10 , 15, 20,25,30,35,40}, and the optimal value, upper bound (UB), lower bound (LB), the number of generated nodes (Nodes), the computational time in second (Time), and the number of unsolved problems (Status). The stopping condition for the BAB algorithm was determined and consider that the problem is unsolved (state is 1), that the BAB algorithm is stopped after a fixed period of time, here after 1800 second (i.e. after 30 minutes). We observed from table (1.1), the heuristic of upper bound is good algorithm, it gives the value for objective function equal to optimal or near optimal value.

Table 1. Table performance optimal, lower bound, upper bound, number of nodes and computational time in second of BAB algorithm.

n	EX	Optimal	UB	ILB	Node	Time	Status
10	1	8	12	8*	54	0.0044	0
	2	40	41	40*	54	0 .0138	0
	3	13	17	13*	32093	4.2245	0
	4	24	25	24*	56	0.0067	0
	5	59	60	59*	54	0.0072	0
	6	14	26	14*	72	0.0095	0
	7	29	42	29*	94	0.0136	0
	8	26	28	26*	54	0.0067	0
	9	26	29	25	1496383	126.3626	0
	10	14	14**	13	1113331	102.4408	0
20	1	5	5**	5*	0	0.005	0
	2	34	37	34	209	0.0279	0
	3	40	46	40*	209	0.0251	0
	4	18	28	18*	8961642	802.5146	0
	5	19	22	19*	6799214	576.8800	0
	6	39	47	37	52914723	1800	1
	7	56	60	56*	77424848	636.2999	0
	8	4	4**	4*	0	0.004	0
	9	77	81	77*	69489	7.5061	0
	10	22	23	22*	209	0.0169	0

30	1	24	24	22	48219967	1800	1
	2	34	43	29	55324069	1800	1
	3	181	185	178	45919155	1800	1
	4	18	18*	17	49189671	1800	1
	5	29	34	29	464	0.0334	0
	6	90	98	90*	464	0.0440	0
	7	105	110	105*	464	0.0516	0
	8	0	0**	0*	0	0.0002	0
	9	129	133	129*	464	0.0450	0
	10	17	17**	16	0	0.0003	0
40	1	63	71	63*	819	0.6789	0
	2	68	69	66	51508209	1800	1
	3	183	191	180	43949822	1800	1
	4	0	0**	0*	0	0.0006	0
	5	28	34	26	6100106	1800	1
	6	9	9**	9*	0	0.0014	0
	7	21	21**	21*	0	0.0004	0
	8	87	95	7*	819	0.2725	0
	9	200	211	199	4417149	1800	1
	10	22	24	17	5059076	1800	1

Optimal = the optimal value obtained by BAB method. UB = upper bound. ILB = initial lower bound. Nodes = the number of generated nodes. Time = Computational time in seconds.

**= The upper bound gives the optimal value.

*= The lower bound gives the optimal value.

Status = $\begin{cases} 1 \; if \; the \; problem \; unsolved \\ 0 \quad if \; the \; problem \; solved \end{cases}$

References

[1] Potts C. N. and Van Wassenhove L. N.,(1991).Single Machine To Minimize Total Late Work. Operations Research 40-3(1991)586-595.

[2] Błażewicz, J. (1984). Scheduling preemptible tasks on parallel processors with information loss. Recherche Technique et. Science Informatiques , 3, 415–420.

[3] Hariri, A. M. A., Potts, C. N., & Van Wassenhove, L. N. (1995). Single machine scheduling to minimize total weighted late work. ORSA Journal on Computing, 7, 232–242. doi:10.1287/ ijoc.7.2.232.

[4] Kethley, R. B., &Alidaee, B. (2002). Single machine scheduling to minimize total weighted late work: A comparison of scheduling rules and search algorithms. Computers & Industrial Engineering, 43, 509–528. doi:10.1016/S0360-8352(02)00123-7

[5] Błażewicz, J., Pesch, E., Sterna, M., & Werner, F. (2005a). The two-machine flow-shop problem with weighted late work criterion and common due date. European Journal of Operational Research, 165, 408–415. doi:10.1016/j.ejor.2004.04.011

[6] Błażewicz, J., Pesch, E., Sterna, M., & Werner, F. (2005b). A comparison of solution procedures for two-machine flow shop scheduling with late work criterion. Computers & Industrial Engineering,49,611–624. doi:10.1016/j.cie.2005.09.001

[7] Błażewicz, J., Pesch, E., Sterna, M., & Werner, F. (2007). A note on the two machine job shop with the weighted late work criterion. Journal of Scheduling, 10,87–95. doi:10.1007/s10951006-0005-5

[8] Błażewicz, J., Pesch, E., Sterna, M., & Werner, F. (2008). Metaheuristic approaches for the two-machine flow-shop problem with weighted late work criterion and common due dates. Computers & Operations Research, 35, 574–599. doi:10.1016/j.cor.2006.03.021

[9]Lin, B. M. T., Lin, F. C., & Lee, R. C. T. (2006). Two-machine flow-shop scheduling to minimize total late work. Engineering Optimization, 38, 501–509. doi:10.1080/03052150500420439

[10] AL Zuwaini M.K,(2006). A comparative Study of Local Search Methods for Some Machine Scheduling Problems with the Usage of Hybridization As A Tool. Ph. D. thesis, University of Baghdad, College of Education (Ibn Al-Haitham), Dep. of Mathematics.

[11] Al-Ayoubi M.G.(2012) A single machine scheduling problem to minimize the sum of total late work and total completion time. Mathematic Dept College of science Al-Mustansiriyah University.

[12] Al_Nuaimi. A.A.(2014). Muiticriteria scheduling: Mathematical, Models, Exact and Approximation algorithms.M.Sc. thesis, Dept. of mathematics, college of science university of AL-Mustansiriya.

[13]Moore, J.M., (1968). An n job one machine algorithm for minimizing the number of late jobs. Management Science 15, 102–109.

[14]Araibi S.M.,(2012).Machine Scheduling Problem to Minimize Two and Three Objectives Function.M.Sc thesis, Dept. of mathematics, college of Education for Pur Sciences, Thi-Qar University.

[15] Husein N.A.,(2012). Machine Scheduling Problem to Minimize Multiple Objective Function., M.Sc. thesis, Dept. of mathematics, college of Education (Ibn AL- Haitham), Baghdad University.

[16] P. Baptiste.,(1999). An O(n4) algorithm for preemptive scheduling of a single machine to minimize the number of late jobs Operations Research letters, 24:175-180, (1999).

[17] Potts, C. N. and L. N. Van Wassenhove,(1992).Approximation Algorithms for Scheduling a Single Machine to Minimize Total Late Work, Operations Research Letters 11(1992)261-266

[18]Abdul-Razaq, T.S., Potts C.N. and Van Wassenhove,(1990). A survey of algorithms for the single machine total weighted tardiness scheduling problem Discrete App. Math. 26(1990) 235-253.

Authors' information

Author (1)

Name:. Mohammed.

Nationality: Iraqi

Address: Iraq/ Thi-Qar/ Al-Nasiriya

Languages: Arabic

Author (2)

Name: Asmaa A.Z.

Nationality: Iraqi

Address: Iraq/ Thi-Qar/ Al-Nasiriya

Languages: Arabic

Education: B.Sc./ Mathematics/ Thi-Qar University/ Thi-Qar/ Iraq/ 2012.

M.Sc./ Mathematics/ Thi-Qar University/ Thi-Qar/ Iraq/ 2016.