Exact algorithm for minimizing the sum of total late work and maximum late work problem

Adawiyah A. Mahmood Al-Nuaimy
Mathematics Department
College of Science
Diyala University

Tariq S. Abdul-Razaq
Mathematics Department
College of Science
Al-Mustansiriyah University

Abstract

This paper presents a branch and bound (BAB) algorithm for minimizing the sum of total late work and maximum late work problem within the single machine context.

Late work is the amount of work executed after a given due date. Branch and bound (BAB) is proposed, two heuristic methods are used to find an upper bound. This BAB proposes a lower bound based on the decomposition property of the bi-criteria problem. Based on results of computational experiments, conclusions are formulated on the efficiency of the BAB algorithm.

Keywords: Late work criterion, branch and bound algorithm, bi-criteria scheduling.

خوارزمية مضبوطة لتصغير مسألة المجموع لعمل متأخر كلي وأعظم عمل متأخر

م. عدوية علي محمود النعيمي

أ.طارق صالح عبد الرزاق

المستخلص

إن هذا البحث يقدم خوارزمية التقيد والتفرع (Branch and bound(BAB)) لمسألة تصغير المجموع لعمل متأخر كلي وأعظم عمل متأخر(The sum of total late work and maximum late work) على ماكنة واحدة.

العمل المتأخر هو مقدار العمل الذي ينفذ بعد وقت مثالي معطى. استخدمنا طريقتين تقريبيتين لإيجاد القيد الأعلى (Upper bound). في خوارزمية التقيد والتفرع يتم إيجاد قيد أدنى (Lower bound) يعتمد على تجزئة المسألة ثنائية المقاييس.

كلمات مفتاحية: مقياس عمل متأخر, خوارزمية التقيد والتفرع , جدولة مقاييس ثنائية.

Introduction

In general, the scheduling problem is defined as a problem of assigning a set of jobs to a set of machines in time under given constraints ([2],[6],[10]). Jobs j (j=1,2,…,n) are mainly characterized by processing times (p_j), due dates (d_j), define expected completion times ($C_j=\sum_{i=1}^{j} p_i$) for particularschedule of jobs. The quality of an assignment, i.e. a schedule, can be evaluated from different points of view, which are represented by different performance measures. Most objective functions based on due dates are regular ones, i.e. non-decreasing with increase in completion times of jobs. This group includes criteria based on lateness ($L_j=C_j-d_j$), tardiness ($T_j=\max\{0,C_j-d_j\}$) or the number of tardy jobs ($U_j=1$, if $C_j>d_j$, otherwise $U_j=0$). The criteria based on earliness ($E_j=\max\{0,d_j-C_j\}$) are non-regular ones (cf. Fig.1).

The late work criterion estimates the quality of a solution on the basis of the duration of late parts of particular jobs. Late work combines the features of two parameters : tardiness and the number of tardy jobs. Formally speaking, in the non-preemptive case the late work parameter for job j in a given schedule is defined as

$V_j=\min\{\max\{0,C_j-d_j\},p_j\}=\min\{T_j,p_j\}$ or, in a more extensive way, as

$$V_j=\begin{cases} 0 & C_j\le d_j \\ C_j-d_j & d_j<C_j<d_j+p_j, \quad j=1,2,\ldots,n \\ p_j & C_j\ge d_j+p_j \end{cases}$$

The parameter V_j was first introduced by Blazewicz[3], who called it "information loss", referring to a possible application of the performance measures based on it. The phrase "late work" was proposed by Potts and Van Wassenhove[11]. Some researchers, e.g. Hochbaum and Shamir[7], use a descriptive name for this schedule parameter-the number of tardy job units.

The relation between late work and other performance measures was established by Blazewicz et al.[4], see fig. (1)

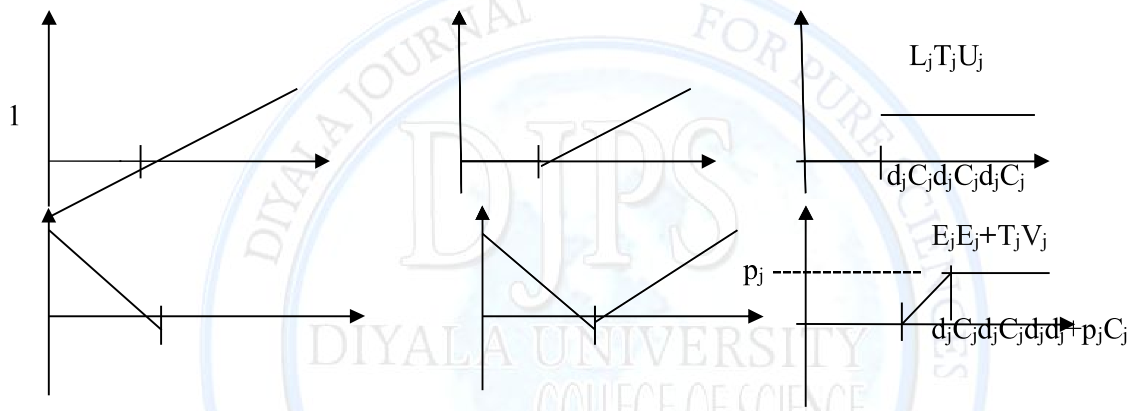

Fig. (1). Schedule parameters based on due dates

Applications of the late work minimization problems arise in control systems ([3],[11]), where the accuracy of control procedures depends on the amount of information provided as their input. Leung [5] pointed out another application of late work scheduling in computerized control systems, where data are collected and processed periodically.

The late work parameter appears to be important in production planning both from the customer's point of view and from the manager's point of view. If the customer orders are interpreted as jobs to be executed, then minimizing the late work is equivalent to minimizing those parts of orders which are not executed on time.

Interesting applications of the late work criteria arise in agriculture, where performance measures based on due-dates are especially useful [1]. Late work criteria can be applied in any situation where a perishable commodity is involved [11].

The organization of this paper is as follows. Section 2 presents the problem formulation. Section 3 provides a general framework of a branch and bound is proposed, incorporating techniques to calculate upper and lower bounds of the criterion value. Section 4 summarizes results of computational experiments and it is followed by a conclusions is given in section 5.

Problem formulation

I. A set of n independent jobs N={1,2,…,n} are available for processing at time zero, job j (j=1,2,…,n) is to be processed without interruption on a single machine that can be handle only one job at a time, requires processing time p_j and due date d_j. For a given schedule σ of thejobs, completion time $C_{\sigma(j)}=\sum_{i=1}^{j} p_{\sigma(i)}$, total latework $\sum_{j=1}^{n} V_{\sigma(j)}$and maximum late work $V_{max}(\sigma)=\max\{V_{\sigma(1)},V_{\sigma(2)},\dots,V_{\sigma(n)}\}$ can be computed where

$$V_{\sigma(j)} = \begin{cases} 0 & \text{if } C_{\sigma(j)} \le d_{\sigma(j)} \\ C_{\sigma(j)} - d_{\sigma(j)} & \text{if } d_{\sigma(j)} < C_{\sigma(j)} < d_{\sigma(j)} + p_{\sigma(j)} \\ P_{\sigma(j)} & \text{if } C_{\sigma(j)} \ge d_{\sigma(j)} + p_{\sigma(j)} \end{cases}$$

The objective is to schedule the jobs so that the criteria $\sum_{j=1}^{n} V_j+V_{max}$is minimized. Note that this problem is NP-hard since the $1//\sum_{j=1}^{n} V_j$ problem is NP-hard.

This problem denoted by (P) and can be stated as follows :

$$\left.\begin{array}{ll}
Z=\min\{\sum V_{\sigma(j)}+V_{max}(\sigma)\} & \\
\sigma\epsilon S & \\
S.t. & \\
C_{\sigma(j)} \geq p_{\sigma(j)} & j=1,2,\ldots,n \\
C_{\sigma(j)}= C_{\sigma(j-1)}+p_{\sigma(j)} & j=1,2,\ldots,n \text{ ------ (P)} \\
V_{\sigma(j)} \leq C_{\sigma(j)}-d_{\sigma(j)} & j=1,2,\ldots,n \\
V_{\sigma(j)} \leq p_{\sigma(j)} & j=1,2,\ldots,n \\
V_{\sigma(j)} \geq 0 & j=1,2,\ldots,n
\end{array}\right\}$$

Where S is the set of all schedules. This problem can be decomposed into two subproblems (P1) and (P2).

Branch and bound (BAB) algorithm

Branch and bound (BAB) algorithm is enumeration technique which can find an optimal solution by systematically examining subsets of feasible solutions. The branch and bound finds s^* by implicit enumeration all s ϵ S through examination of smaller subsets of the set of feasible solutions S.These subsets can be treated as sets of solutions of corresponding subproblems of the original problem.

The BAB algorithm is determined by the following procedures :

1.The branching procedure :

This describes the method to partition subsets of possible solutions. These subsets can be treated as a set of solutions of corresponding subproblems of the original problem.

2.The bounding procedure :

This indicates how to calculate a lower bound (LB) on the optimal solution value for each subproblem generated in the branching process.

3.The search strategy :

This strategy describes the method of choosing a node of the search tree to branch from it, we usually branch from a node with the smallest lower bound (LB) among the recently created nodes.

A general description of the BAB algorithmwill now be given. The set of all possible schedules is divided up into disjoint subsets, each of which may contain more than one possible schedule. For each subset we calculate a (LB) which is the cost of the sequenced jobs (depending on the objective function) and the cost of the unsequenced jobs (depending on the derived LB). If the (LB) of this subset is greater than or equal to the upper bound (UB), then this subset is ignored (this upper bound UB is the value for a trial solution (schedule). At the beginning the trial solution is obtained by using a heuristic method) since any subset with value less than UB can only exist in the remaining subsets. These remaining subsets have to be considered one at a time.

One of these subsets is chosen according to some search strategy, from which to branch. This subset is then divided (as above) into smaller disjoint subsets. As soon as one of these subsets contains one element only, a complete sequence of the jobs should exist, this sequence is evaluated and if its value is less than the current upper bound UB, this UB is reset to take that value. The procedure is then repeated until all subsets (nodes) have been considered. The upper bound (UB) at the end of this BAB procedure is the optimal solution for the particular problem.

Upper bound (UB)

The upper bound of the criterion value (UB) is used by the BAB algorithm for truncating search tree branches, which do not lead to an optimal solution. During the search, UB is equal to the criterion value for the best solution constructed by BAB so far. At the beginning of the solution process, the initial upper bound is determined by the problem under consideration.

Two heuristic methods which are proposed and applied once at the root node of the (BAB) search tree to find an upper bound (UB) on the minimum value of ($\sum_{j=1}^{n} V_j+V_{max}$).

The first heuristic method (UB1) is obtained by the earliest due date (EDD) rule, that is sequencing the jobs in non-decreasing order of their due date (d_j), j=1,2,…,n. For the resulting schedule compute UB1=$\sum_{j=1}^{n} V_j+V_{max}$.

The second heuristic method (UB2) is obtained by Lawler's algorithm, which solves the $1/prec/f_{max}$problem or $1//f_{max}$problem where $f_{max}\in \{C_{max},L_{max} , T_{max} ,V_{max}\}$ [10].

Lower bound (LB)

Deriving a lower bound (LB) for the bi-criteria problem (P) is based on decomposing the problem into two subproblems (P1) and (P2), then the lower bound is the sum of the lower bound of the subproblem (P1) and the minimum value of the subproblem (P2).

$$Z1 = \min_{\sigma\in S} \sum_{j=1}^{n} V_{\sigma(j)}$$

S.t.

$$V_{\sigma(j)} = \begin{cases} 0 & \text{if } C_{\sigma(j)} \le d_{\sigma(j)} \quad j=1,2,\dots,n \\ C_{\sigma(j)} - d_{\sigma(j)} & \text{if } d_{\sigma(j)}<C_{\sigma(j)}<d_{\sigma(j)}+p_{\sigma(j)} \quad j=1,2,\dots,n \\ P_{\sigma(j)} & \text{if } C_{\sigma(j)} \ge d_{\sigma(j)}+p_{\sigma(j)} \quad j=1,2,\dots,n \end{cases} \qquad (P1)$$

$$\left.\begin{array}{lll} Z2 = \text{Min } V_{max}(\sigma) & & \\ \sigma\epsilon S & & \\ \text{S.t.} & & \\ V_{\sigma(j)} \leq C_{\sigma(j)} - d_{\sigma(j)} & j=1,2,\ldots,n & \text{(P2)} \\ V_{\sigma(j)} \leq p_{\sigma(j)} & j=1,2,\ldots,n & \\ V_{\sigma(j)} \geq 0 & j=1,2,\ldots,n & \end{array}\right\}$$

This decomposition has the following properties;

first (P1) and (P2) have simpler structure than the bi-criteria problem, a lower bound (V1) can be obtained for (P1) by setting V1= T_{max}(EDD) since $T_{max}(EDD) \leq \sum_{j=1}^{n} V_j$. Second it is easy to solve optimality for (P2) by applying Lawler's algorithm

$V2 = V_{max}$(Lawler).

Thus a lower bound LB = V1+V2.

Theorem[9]

If V1, V2 and Z are the minimum objective function values of (P1), (P2) and (P) respectively, then V1+V2 ≤ Z. □

Computational results

The BAB algorithm is tested by coding it in Matlab R2009b and running on a personal computer hp with Ram 2.50 GB. Test problems are generated as follows : for each job j, an integer processing time $p_j$is generated from the discrete uniform distribution [1,10]. Also, for each job j, an integer due date is generated from the discrete uniform distribution [P(1-TF-RDD/2),P(1-TF+RDD/2)], where $P = \sum_{j=1}^{n} p_j$, depending on the relative range of due date (RDD) and on the average tardiness factor (TF). For both parameters, the values

0.2,0.4,0.6,0.8,1.0 are considered. For each selected value of n, two problems are generated for each of the five values of parameters producing 10 problems for each value of n, where the number of jobs n=5,10,15,20.

The following tables (4.1),…,(4.4) give the comparative of computational results (optimal, first upper bound (UB1), second upper bound (UB2), lower bound (LB) and the time (in seconds) which is required for the BAB algorithm) for the problem (P). Whenever a problem could not be solved to optimality within the time limit of 1800 second, computation is abandoned for that problem. In all of these tables we have :

*: indicates that the problem has an optimal solution equal to the heuristic value UB = min{UB1,UB2}.

**: indicates that the problem has an optimal solution equal to the lower bound (LB).

Nodes = the number of generated nodes.

Status = { 1 if the example is solved; 0 otherwise }

Table (4.1) : The performance of lower bound, upper bounds and computational time of BAB algorithm for n=5.

EX	Optimal	UB1	UB2	LB	Nodes	Time	Status
1	14	14*	14*	8	95	0.0704	1
2	22	26	22*	22**	0	0.0316	1
3	19	28	19*	19**	0	0.0316	1

4	27	29	28	27**	32	0.0889	1
5	12	14	14	11	117	0.0939	1
6	10	15	10*	10**	0	0.0318	1
7	21	24	21*	21**	0	0.0346	1
8	8	11	8*	8**	0	0.0314	1
9	29	38	29*	25	142	0.0949	1
10	21	28	21*	21**	0	0.0749	1

Table (4.2) : The performance of lower bound, upper bounds and computational time of BAB algorithm for n=10.

EX	Optimal	UB1	UB2	LB	Nodes	Time	Status
1	53	57	53*	52	778798	31.3259	1
2	27	37	28	20	755602	34.8771	1
3	41	50	41*	40	178123	6.7505	1
4	26	37	2.6*	23	85646	3.2578	1
5	52	61	52*	51	373343	14.2586	1
6	27	39	31	19	574797	22.8078	1
7	43	57	51	32	698113	32.5934	1
8	23	32	23*	16	258783	10.5334	1

9	11	11*	11*	8	843	0.1361	1
10	3	3*	3*	2	285	0.1084	1

II. Table (4.3) : The performance of lower bound, upper bounds and computational time of BAB algorithm for n=15.

EX	Optimal	UB1	UB2	LB	Nodes	Time	Status
1	10	10*	10*	8	431134	25.7285	1
2	38	85	38*	37	48993286	1800.003	0
3	16	17	16*	10	19645598	1090.003	1
4	23	43	23*	12	27875746	1273.003	1
5	25	32	25*	16	30622579	1800.003	0
6	17	45	17*	16	442042	15.8575	1
7	49	71	57	42	32565215	1800.003	0
8	92	98	92*	92**	0	0.0780	1
9	14	45	14*	14**	0	0.03328	1
10	24	38	24*	24**	0	0.0323	1

Table (4.4) : The performance of lower bound, upper bounds and computational time of BAB algorithm for n=20.

EX	Optimal	UB1	UB2	LB	Nodes	Time	Status
1	55	103	55*	55**	0	0.0807	1
2	53	72	55	30	28560125	1800.003	0
3	78	119	108	76	35613973	1800.003	0
4	17	34	17*	10	43476785	1800.003	0
5	47	102	47*	42	39840633	1800.003	0
6	44	52	44*	44**	0	0.0707	1
7	66	112	66*	66**	0	0.0808	1
8	65	82	65*	65**	0	0.0807	1
9	68	116	68*	68**	0	0.0809	1
10	35	58	35*	35**	0	0.0708	1

Table (4.5) summary of the tables (4.1) to (4.4) of BAB algorithm

n	Average nodes	Average time	Unsolved problems
5	38.6	0.0584	0
10	370433.3	15.6649	0
15	16057560	780.474458	3
20	14749151.6	720.04766	4

Table (4.5) is the summary of the previous tables (4.1) to (4.4) that shows the average of nodes, the average computational time and the unsolved problems for the 10 problems for each n, where nϵ\{5,10,15,20\}, increase whenever n increases.

Conclusions

In this paper a branch and bound (BAB) algorithm is proposed to find an optimal solution for the problem of minimizing a bi-criteria $\sum_{j=1}^{n} V_j + V_{max}$. A computational experiment for the branch and bound (BAB) algorithm on a large set of test problems are given.

The main conclusion to be drawn for our computational results is that the second upper bound (UB2) is more effective than the first upper bound (UB1) for our problem.

An interesting future research topic would involve experimentation with the approximation algorithms for large n and experimentation with the following bi-criteria problems :

1.1//Lex($\sum V_i$, V_{max}).

2.1//F($\sum V_i$, V_{max}).

References

1. Alaminana M,Escudero LF,Landete M,Monge JF,Rabasa A,Sanchez-Soriano J. WISCHE;A DSS for water irrigation scheduling. Omega 2010; 38 (6): 492-500.
2. Blazewicz J,Ecker K, Pesch E, Schmidt G, WeglarzJ.Handbook on scheduling. From theory to applications. Berlin-Heidelberg-NewYork : Springer; 2007.
3. Blazewicz J. Scheduling preemptible tasks on parallel processorswith information loss. Technique et Science Informatiques 1984; 3 (6):415-20.
4. Blazewicz J, Pesch E, Sterna M, Werner F. Total late work criteria for shop scheduling problems. In : Inderfurth K, Schwoediauer G, Domschke W, Juhnke F, Kleinschmidt P, Waescher G. editors. Operations Research Proceedings 1999.Berlin : Springer; 2000. P.354 – 9.
5. Blazewicz J, Pesch E, Sterna M, Werner F. Open shop scheduling problems with late work criteria. Discrete Applied Mathematics 2004; 134 (1) : 1-24.
6. Brucker P. Scheduling algorithms. Berlin-Heidelberg-New York : Springer; 2007.
7. Hochbaum DS, Shamir R. Minimizing the number of tardy job units under release time constraints. Discrete Applied Mathematics 1990; 28 (1) : 45-57.
8. Leung JY-T. Minimizing total weighted error for imprecise computation tasks and related problems. In : JY-T Leung. editor. Handbook of scheduling : algorithms, models and performance analysis. Boca Raton : CRC Press; 2004. P.34. 1-16.
9. Mahmood A. A.," Solution procedures for scheduling job families with setups and due dates " M.Sc. Thesis, Univ. of Al-Mustansiriyah, College of Science, Dept. of Mathematics (2001).
10. Pinedo M. Scheduling : theory, algorithms and systems. New York : Springer; 2008.
11. Potts CN. Van Wassenhove LN. Single machine scheduling to minimize total late work. Operations Research 1991; 40 (3) : 586-95.
12. Sterna M. Problems and algorithms in non-classical shop scheduling. Poznan : Scientific Publishers of the Polish Academy of Sciences; 2000.