Optimal Solution for Simultaneous Multicriteria Problem

Adawiya A. Mahmood Al-Nuaimi

Mathematics Department - College of Science - Diyala University

Received 18 March 2015; Accepted 29 April 2015

Abstract

This paper considers a branch and bound (BAB) algorithm for simultaneous multicriteria problem of minimizing the sum of the three criteria of total completion time, maximum tardiness and maximum late work within the single machine context.Late work is the amount of work executed after a given due date. Heuristic method was used to find an upper bound. This BAB proposes a lower bound based on the decomposition property of the multicriteria problem. Based on results of computational experiments, conclusions are presented on the efficiency of the BAB algorithm.

Keywords: Multicriteria scheduling, late work criterion, branch and bound algorithm, single machine.

حل أمثل لمسألة متعددة المقاييس تحدث في وقت واحد

عدوية علي محمود النعيمي

قسم الرياضيات - كلية العلوم - جامعة ديالى

الخلاصة

إن هذا البحث يقدم خوارزمية التفرع والتقيد ((Branch and bound(BAB) لمسألة متعددة المقاييس تحدث في وقت واحد لتقليل المجموع للمقاييس الثلاثة لوقت الإتمام الكلي($\sum C_j$) ، أعظم تأخير لا سالب(T_{max}) وأعظم تأخير لوحدات عمل متأخر(V_{max}) على ماكنة واحدة. العمل المتأخر هو مقدار العمل الذي يُنفذ بعد وقت مثالي معطى. استُخدمت طريقة تقريبية لإيجاد القيد الأعلى (Upper bound). في خوارزمية التفرع والتقيد يتم إيجاد قيد أدنى(Lower bound) يعتمد على تجزئة المسألة متعددة المقاييس. بالاعتماد على نتائج التجارب الحسابية قُدمت استنتاجات حول كفاءة خوارزمية التفرع والتقيد(BAB).

كلمات مفتاحية: جدولة متعددة المقاييس، مقياس عمل متأخر، خوارزمية التفرع والتقيد، ماكنة واحدة.

Introduction

The machine scheduling problem plays important role in manufacturing and production systems as well as in information systems. The basic scheduling problem can be described as finding for each task (job), an execution interval on one or more than one machine such that the resulting solution, which is called a schedule minimizes the given objective function ([4], [3]). The jobs j (j=1,2,…,n) require processing times (p_j), due dates (d_j), define completion times ($C_j = \sum_{i=1}^{j} p_i$) for particular schedule of jobs.

For many years, scheduling researchers focused on single regular performance measures that are non-decreasing in job completion time. Typically, each criterion has been studied separately, even though most real life scheduling problems involve multiple criteria [7]. However, few studies considered multiple criteria together. Most multicriteria scheduling problems are NP-hard in nature [1], where NP means Non-Deterministic Polynomial time.

The late work criterion for job j in a given schedule is defined as follows:

$$V_j = \min\{\max\{0, C_j - d_j\}, p_j\}.$$

The problems that contain the late work criterion were studied by Potts, Van Wassenhove [9] and Blazewicz [2].

Some applications of the late work problems occur in control procedures and production planning ([2],[9]).

The organization of this paper is as follows. Section 2 presents simultaneous optimization of multicriteria. 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 conclusions are given in section 5.

Simultaneous minimization of multicriteria

The three performance criteria were indicated by the cost function f_i (i=1,2,3) are transferred into single composite objective function F: S→R, where S is the set of all schedules. We restrict F to a linear composition of the performance criteria f_i . The optimization problem for the performance criteria of total completion time ($\sum C_j$), maximum tardiness (T_{max}) and maximum late work (V_{max}) is denoted by $1//\sum C_j + T_{max} + V_{max}$ and called it (P). The problem (P) from the class of simultaneous optimization can be modeled as follows:

$$\left.\begin{array}{ll}
\mathbf{Z = Min\{ \sum_{j=1}^{n} C_{\sigma(j)} + T_{max}(\sigma) + V_{max}(\sigma) \}} & \\
\quad \sigma \in S & \\
\quad \text{Subject to} & \\
C_{\sigma(1)} = p_{\sigma(1)} & \\
C_{\sigma(j)} = C_{\sigma(j-1)} + p_{\sigma(j)} & j=2,3,\ldots,n \\
T_{\sigma(j)} \geq C_{\sigma(j)} - d_{\sigma(j)} & \forall j,\ j = 1,2,\ldots,n \\
T_{\sigma(j)} \geq 0 & \\
V_{\sigma(j)} \leq C_{\sigma(j)} - d_{\sigma(j)} & \\
V_{\sigma(j)} \leq p_{\sigma(j)} & \\
V_{\sigma(j)} \geq 0 &
\end{array}\right\} \text{-------(P)}$$

Where σ is a given schedule of the jobs j, (j=1,…,n) and S is the set of all schedules.

The goal in problem (P) is to find a processing order of the jobs on a single machine to minimize the sum of total completion time, maximum tardiness and the maximum late work (i.e.$1//\sum C_j + T_{max} + V_{max}$). The problem (P) is decomposed into three subproblems (S1), (S2) and (S3) with a simpler structure as follows:

$$\left.\begin{array}{ll}
Z1 = \underset{\sigma \in S}{Min}\{\sum_{j=1}^{n} C_{\sigma(j)}\} & \\
\text{S.t.} & \\
C_{\sigma(1)} = p_{\sigma(1)} & \\
C_{\sigma(j)} = C_{\sigma(j-1)} + p_{\sigma(j)} & j=2,3,\ldots,n
\end{array}\right\} \ldots(S1)$$

The sub problem (S1) was solved by the SPT (Shortest Processing Time) rule, that is, sequencing the jobs in non-decreasing order of their processing times p_j.

$$\left.\begin{aligned} &\mathbf{Z2}= \min_{\sigma\in S}\{ T_{max}(\sigma)\} \\ &\textbf{S.t.} \\ &T_{\sigma(j)} \geq C_{\sigma(j)} - d_{\sigma(j)} \qquad \forall\, j,\ j = 1,2,\ldots,n \\ &T_{\sigma(j)} \geq 0 \end{aligned}\right\} \qquad \ldots(S2)$$

The sub problem (S2) was solved by the EDD (Earliest Due Date) rule, that is, sequencing the jobs in non-decreasing order of their due dates d_j.

$$\left.\begin{aligned} &\mathbf{Z3}= \min_{\sigma\in S}\{ V_{max}(\sigma)\} \\ &\textbf{S.t.} \\ &V_{\sigma(j)} \leq C_{\sigma(j)} - d_{\sigma(j)} \qquad \forall\, j,\ j = 1,2,\ldots,n \\ &V_{\sigma(j)} \leq p_{\sigma(j)} \\ &V_{\sigma(j)} \geq 0 \end{aligned}\right\} \qquad \ldots(S3)$$

The sub problem (S3) is solved by Lawler's algorithm (LA) [8].

Branch and Bound (BAB) algorithm

Branch and Bound (BAB) algorithm is a general procedure for solving many types of combinatorial optimization problem. BAB method is the most widely solution technique that is used in scheduling [5].This method is the typical example of the implicit enumeration approach, which can find an optimal solution by systematically examining subsets of feasible solutions. The procedure is usually described by means of search tree with nodes that correspond to these subsets. From each node for a partially complete solution there grows a number of new branches which replace the original one by set of new smaller problems that are mutually exclusive. There are two common types of branching: the forward branching, that is, the jobs are sequenced one by one from the beginning and the backward branching , that is, the jobs are sequenced one by one from the end. To minimize an objective function Z, for a particular scheduling problem, the BAB method successively partitions the problem into subsets by using a branching procedure and computes bound by using a lower bounding procedure. By these procedures the subsets which are found not to include any optimal solution are excluded. This leads to at least one optimal solution. The bounding procedure is used to calculate a lower bound (LB) on the solution to each generated subproblem. For each node, LB is calculated which is the cost of the scheduling jobs (depending on the objective function) and the cost of the unscheduled jobs (depending on the derived lower bound). If this node has a value LB greater than or equal to the upper bound (UB) the upper bound is usually defined as the minimum of the values of all feasible solutions currently found, then this node is dominated and one of the remaining nodes that has the least LB is chosen. 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 repeated until all nodes have been considered, i.e., in the search tree $LB \geq UB$ for all nodes. A feasible solution with this UB is an optimal solution for this problem. To solve the problem (P) by using the branch and bound (BAB) algorithm, lower and upper bounds were calculated by using the following techniques:

Heuristic technique

At the beginning of the solution process, the initial upper bound UB is determined for the problem under consideration and used in BAB algorithm.

The heuristic method gives UB is obtained by the (SPT) rule, that is sequencing the jobs in non-decreasing order of their processing times (p_j), j=1,2,…,n. For the resulting schedule compute $UB=\sum_{j=1}^{n} C_j + T_{max} + V_{max}$.

Decomposing technique

Decomposing the problem (P) into three subproblems (S1), (S2) and (S3) as mentioned above is used to find the lower bound (LB). Then the lower bound (LB)of the problem (P) is the sum of minimum values of the subproblems (S1), (S2) and (S3).It is easily to solve optimality for (S1), (S2) and (S3) by SPT rule, EDD rule and Lawler's algorithm (LA) respectively since the decomposition has simpler structure than (P).

Let Z1, Z2 and Z3 be the minimum values of (S1), (S2) and (S3), then applying the following theorem to get a lower bound for (P).

Theorem (3.2) [6]

If Z1, Z2, Z3 and Z are the minimum objective function values of (S1), (S2), (S3) and (P) respectively then Z1+Z2+Z3 $\leq$ Z.■

By applying theorem (3.2) a lower bound (LB) for the problem (P) is given by

LB=Z1+Z2+Z3.

Test problems with computational experiments

Test problems were generated as follows:

For each job j, an integer processing time p_j is generated from the discrete uniform distribution in the interval [1,10]. Also, for each job j, an integer due date d_j 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 [9]. 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 BAB algorithm was tested on the problem (P) by coding it in Matlab R2009b and running on a personal computer hp with Ram 2.50 GB. The computational results (optimal, upper bound (UB), lower bound (LB) and the time (in seconds) which is required for the BAB algorithm) for the problem (P) are given in the tables (4.1),…,(4.4). Computation is abandoned for the problem, whenever that problem could not be solved to optimality within the time limit of 1800 second. All of these tables have:

*****: indicates that the optimal solution equal to the UB value.

******: indicates that the optimal solution equal to the LB value.

Nodes = the generated nodes number.

$$\text{Status} = \begin{cases} 1, & \text{if the example is solved} \\ 0, & \text{otherwise} \end{cases}$$

Table (4.1): The results of LB, UB and computational time of BAB algorithm for n=5.

EX	Optimal	UB	LB	Nodes	Time	Status
1	78	78*	78**	0	0.0953	1
2	81	81*	78	205	0.1514	1
3	82	82*	80	205	0.1508	1
4	85	85*	84	197	0.1531	1
5	61	61*	58	205	0.1538	1
6	67	67*	63	201	0.1552	1
7	82	82*	79	205	0.1525	1
8	53	56	50	205	0.1525	1
9	125	133	120	205	0.1619	1
10	76	76*	74	205	0.1517	1

Table (4.2): The results of LB, UB and computational time of BAB algorithm for n=10.

EX	Optimal	UB	LB	Nodes	Time	Status
1	282	282*	279	5657326	506.6223	1
2	206	212	193	4122810	385.2171	1
3	221	222	216	4670291	445.2931	1
4	284	298	278	5198020	501.9450	1
5	285	285*	282	5096528	493.1460	1
6	209	213	192	3554677	344.2807	1
7	262	262*	247	5456751	504.6112	1
8	185	185*	167	3445637	324.2903	1
9	283	283*	283**	0	0.0967	1
10	246	246*	246**	0	0.0932	1

Table (4.3): The results of LB, UB and computational time of BAB algorithm for n=15.

EX	Optimal	UB	LB	Nodes	Time	Status
1	179	185	167	14513069	385.1171	1
2	209	215	181	4374962	553.1596	1
3	75	86	39	58992398	1800.003	0
4	35	35*	24	28756475	1382.003	1
5	48	48*	48**	0	0.0870	1
6	78	78*	78**	0	0.0790	1
7	267	287	234	9623513	165.4119	1
8	94	94*	94**	0	0.0963	1
9	169	169*	166	15328643	493.7634	1
10	89	98	42	35246319	1800.003	0

Table (4.4): The results of LB, UB and computational time of BAB algorithm for n=20.

EX	Optimal	UB	LB	Nodes	Time	Status
1	266	266*	241	19974588	1491.003	1
2	390	390*	390**	0	0.0709	1
3	285	285*	285**	0	0.0806	1
4	199	201	74	46531798	1800.003	0
5	432	432*	221	49784605	1800.003	0
6	388	388*	274	49968714	1800.003	0
7	250	250*	235	19895437	1482.003	1
8	556	558	420	54894719	1800.003	0
9	420	420*	420**	0	0.0709	1
10	260	260*	260**	0	0.0908	1

Table (4.5): Averages of nodes and computational time.

n	Average nodes	Average time	Unsolved problems
5	183.3	0.14782	0
10	3720204	350.55956	0
15	16683538	657.97333	2
20	24104986	1017.3332	4

Table (4.5) shows the average number of nodes, the average computational time in seconds and the unsolved problems for the 10 problems for each n=5,10,15,20. It is clear from table (4.5) that whenever n increases, the number of nodes and the computational time increase.

Conclusions

This paper proposes the Branch and Bound (BAB) algorithm to solve the multicriteria scheduling problem $\sum C_j + T_{max} + V_{max}$. The complexity of scheduling problem increases whenever the number of criteria increases. The Branch and Bound (BAB) algorithm was applied on a large set of test problems. The computational results show that the upper bound (UB) is effective. A future research topic includes the application of approximation algorithms with the multicriteria scheduling problems.

References

1. Akande S., Oluleye A. E. and Oyetunji E. O., "Reducibility of some multicriteria scheduling problems to bicriteria scheduling problems", International Conference on Industrial Engineering and Operations Management 7(9), 642-651 (2014).

2. Blazewicz J., "Scheduling preemptible tasks on parallel processors with information loss", Technique et Science Informatiques 3(6): 415-20 (1984).

3. Brucker P., "Scheduling algorithms", Berlin-Heidelberg-New York: Springer (2007).

4. Hoogeveen H., "Invited review of multicriteria scheduling", European Journal of Operational Research 167, 592-623 (2005).

5. Lomnicki Z., "A branch and bound algorithm for the exact solution of the three machine scheduling problem", Oper. Res. 16, 89-100 (1965).

6. Mahmood A. A., "Solution procedures for scheduling job families with setups and due dates", M.Sc. Thesis, University of Al-Mustansiriyah, College of Science, Dept. of Mathematics (2001).

7. Nagar A., Haddock J. and Heragu S., "Multiple and bi–criteria scheduling: A literature survey", European Journal of Operational Research 81, 88-104 (1995).

8. Pinedo M. Scheduling : theory, algorithms and systems. New York : Springer; 2008.

9. Potts C. N. and Van Wassenhove L. N., "Single machine scheduling to minimize total late work", Operations Research 40(3):586-595(1991).