Modified Algorithm for Scheduling Problem With Efficient Solutions

Ayad M. Ramadan
College of Science-University of Sulaimani

Received:٢٨ /6/2009, Accepted: 25/1/2010

Abstract

The problem of scheduling n jobs on a single machine is considered, where the jobs are divided into two classes and a machine set up is necessary between jobs of different classes. Jobs i (i= 1,…, n) becomes available for processing at time zero, requires a positive processing time P_i. Disjoint subsets N_1 and N_2 define the partition of jobs into two classes. If two jobs in the same class are sequenced in adjacent positions, then no set up time between these jobs in necessary. We address the bicriterion (multi objective) scheduling problem, the two criteria are the minimization of flow time ($\sum_{i \in N} c_i$) and the minimization maximum Tardiness ($T_{\max}$). We characterized the set of all efficient points and the optimal solution. A modified algorithm presented to find efficient solutions for the problem with set up times. A relation found between number of efficient solutions and range of 'tardiness of shortest processing time (T_{SPT}), tardiness of early due date (T_{EDD})'. This algorithm treats with a case that the set up time in $SPT-$rule is in increasing order. A counter example presented to show that the algorithm will fail if the set up time in $SPT-$rule is in decreasing order. Our task is to present the decision makers with all possible solutions and let them make the final selection. The decision maker has two objectives in mind ($\sum_{i \in N} c_i$) , ($T_{\max}$) and some solutions (efficient), we will choose the best one from the efficient solutions depending on his experiences.

Introduction

In the industrial context, scheduling problems are related to manufacturing resource planning (Rocha et al., 2008).There are many researches considering this type, but few machines or sequence-dependent setups. In scheduling one situation where benefits may result from batching occurs when machines require set-up if they are to process jobs that have different characteristics. The set-up may reflect the need to change a tool or to clean the machine, so that no set-up is required for a job if it belongs to the same family of the previously processed job. Therefore, preemptive scheduling problems are those in which the processing of a job can be temporarily interrupted (Potts & Mikhail, 2000). Many practical scheduling problems involve processing several families of related jobs on common facilities, where a set-up time is incurred whenever there is a switch from processing a job in one family to a job in another family.

For example, consider a mechanical parts manufacturing environment in which jobs have to be sequenced for processing on a multi-tool machine (Crauwels et al., 2005).

There are different definitions of the notion of optimal solutions of a multi objective combinatorial optimization (MOCO) problem (Steiner & Radzk, 2008). Efficiency also called Pareto optimality is the most common one. An efficient solution is one such that there is no other solution which is better on all objectives. First attempt in scheduling to find efficient solutions for a problem was by (Wassenhove & Gelders, 1980), until now and according to our knowledge there is no many attempts to study efficient solutions in scheduling, because we must treated with different objectives without having any additional information about relative importance objectives. In general optimization there are papers treated with efficient solutions, the structure of efficient sets in convex optimization (Ward, 1989). Efficient solutions in mathematical programming
(Lowe et al., 1984). Efficient solutions based on genetic algorithms (Bischoff & Klamroth, 2007). Van Wassenhove proposed an algorithm (Van Wassenhove and Gelders, 1980) to find all efficient solutions for problem (1).

$$\sum_{i \in N} c_i \text{ and } T_{\max} \qquad \ldots (1)$$

We consider the problem of sequencing n jobs on a single machine.
The objective is to find efficient solutions for (2) with set-up times.

$$1/\, S_f \,/ \sum_{i \in N} c_i \text{ and } T_{\max} \qquad \ldots (2)$$

where, (1) is one machine scheduling and (S_f) is set up times.

Formulation of the problem

We are given two families of jobs $J^{(1)} = \{j_1^{(1)}, \ldots, j_{n_1}^{(1)}\}$ and $J^{(2)} = \{j_1^{(2)}, \ldots, j_{n_2}^{(2)}\}$ (Yuan et al., 2005) to be processed in a single machine. Let $x \in \{1,2\}$ be given. The processing time on a job $j_i^{(x)} \in J^{(x)}$ is denoted by $P_i^{(x)}$, and each job has a due date $d_i^{(x)}$. For a given schedule π for the jobs $J^{(1)} \cup J^{(2)}$, we use $C_i^{(x)}(\pi)$ to denote the completion time of a job $j_i^{(x)} \in J^{(x)}$. The lateness of a job $j_i^{(x)} \in J^{(x)}$ under π is denoted by $L_i^{(x)}(\pi)$. The maximum lateness of the job in $J^{(x)}$ under π denoted by $L_{\max}^{(x)}(\pi)$. $f^{(x)}(\pi)$ is used to denote the objective of the jobs in $J^{(x)}$ under π (Shabtay & Steiner, 2007).

In this paper we assume $f^{(x)} \in \{\sum c_i^{(x)}, T_{\max}^{(x)}\}$. The objective of the considered problem is to find a schedule π for the jobs $J^{(1)} \cup J^{(2)}$ such that $f(\pi)$ is small as possible, where

$$f(\pi) = 1/2f / \sum_{i \in N} c_i^{(x)} + T_{\max}^{(x)} \qquad ...(3)$$

and $2f$ means that there are 2 families.

Scheduling two job classes

To find an optimal solution for the problem (3), find efficient solutions for (2). Number of efficient solutions is:

$C_r^n = (n!) \,/\, (r!(n-r)!)$, where n is number of jobs and r is number of families ($r = 2$). Let δ_1 be a set up time from f_1 to f_2 and δ_2 be a set up time from f_2 to f_1.

Corollary:

SPT –rule is one of the feasible solutions, where SPT –rule is: order the jobs in non-decreasing order of P_i.

We have two cases in SPT –rule.

(a) $\delta_1 \le \delta_2$.

(b) $\delta_1 > \delta_2$.

Algorithm: case (a)

This algorithm is modified and depends on the algorithm in (Van Wassenhove & Gelders, 1980).

Step 1: Order the jobs in SPT –rule: if $\delta_1 > \delta_2$ (stop).

Step 2: Find $\sum_{i \in N} c_i$ and $T_{\max}$, $\Delta = T_{\max}(SPT) - 1$.

Step 3: $D_i = d_i + \Delta$, use modified smith algorithm, such that $\sum P_i = \sum P_i + \min\{\delta_1, \delta_2\}$, with the precedence between jobs. If a sequence exists, then it is efficient. Else go to step (5).

Step 4: If we get a sequence (π) and $T(\pi) = T(EDD)$, where EDD is early due date, then it is efficient Go to step (5).
Else it is efficient. Go to step (2).

Step 5: stop.

An example to explain set-up times, consider the problem with 4-jobs

That is partitioned into 2 families defined by $\{1,2\}$ and $\{3,4\}$, respectively. Let $\delta_1 = 3$ and $\delta_2 = 4$. The processing times are 5, 7, 10 and 3, and the due dates are 10, 25, 15 and 20, respectively.

i	1	2	3	4
p_i	5	7	10	3
d_i	10	25	15	20

There are two families $f_1 = \{1,2\}$ and $f_2 = \{3,4\}$, from f_1 to f_2, $\delta_1 = 3$ and from f_2 to f_1, $\delta_2 = 4$.

Example:

i	1	2	3	4
P_i	5	2	3	4
d_i	6	3	12	8

Such that $f_1 = \{2,1\}$, $f_2 = \{3,4\}$ with $\delta_1 = 2$, $\delta_2 = 3$.

Solution:

SPT –rule is (2, 3, 4, 1), with $T_{\max} = 10$, $\sum_{i \in N} c_i = 36$.

$\Delta = T_{\max} - 1 = 9$.

$D_i = d_i + \Delta = (15,12,21,17)$.

$\sum_{i \in N} P_i + \min\{2,3\} = 16$, so which of the jobs in (15, 12, 21, 17) ≥ 16. Clearly job (3) and job (4) satisfy the inequality, we choose job 4 because $P_4 > P_3$, arrange job (4) in position K (last). $(K = 4)$.

Now which (15, 12, 21) ≥ 12, job (1) satisfies the inequality, arrange it in position (3).

Which (12, 21) ≥ 7, job (3) satisfies the inequality, arrange it in position (2) and job (2) in position (1). So we get the sequence (2, 3, 1, 4) with $\sum_{i \in N} c_i = 37$, $T_{\max} = 8$. So $\pi = (37, 8)$. It is the second efficient solution.

Iteration 2: $\Delta = 8 - 1 = 7$, $D_i = (13,10,19,15)$ ≥ 16, we get the sequence (2, 4, 1, 3) with (39, 7).

Iteration 3: We get (2, 1, 4, 3) with (38, 5). So the efficient solutions are:

Sequence	$(\sum_{i \in N} c_i , T_{\max})$
(2, 3, 4, 1)	(36, 9)
(2, 3, 1, 4)	(37, 8)
(2, 4, 1, 3)	(39, 7)
(2, 1, 4, 3)	(38, 5) stop $T(EDD) = 5$.

Note:

If there is no set up times, the problem is solved by the algorithm (Van Wassenhove & Gelders, 1980).

Counter example:

The algorithm fails to find efficient solutions for the problem if $\delta_1 > \delta_2$, here a counter example for this case.

i	1	2	3	4
P_i	3	1	5	2
d_i	7	2	3	10

with $\{1,2\},\{3,4\}$ and $\delta_1 = 5$, $\delta_2 = 3$.

SPT –sequence is (2, 4, 1, 3) with $\sum_{i \in N} c_i = 36$, $T_{\max} = 13$, where (4, 2, 1, 3) is another sequence with $\sum_{i \in N} c_i = 31$, $T_{\max} = 11$.

Computer results

For each n= 10, 20,…, 50, we gave 10 examples. The algorithm was tested by coding them in FORTRAN 2003 .Running it on PC IV-1-8 GHz processor and 1GB RAM. Data were generated as follows:

For job i (i= 1, 2, …, n) an integer processing time P_i generated from the uniform distribution [0,100], an integer due-date d_i is generated from the uniform distribution $[0, P_i]$. Table (1) gives results of the algorithm with (n=10) for 10 examples. For (n=20, 30, 40, 50) the results are in APPENDIX A. We found number of efficient points, range between sum of completion times and maximum tardiness and the range between tardiness of (SPT-rule) and (EDD-rule) for each n.

Table (1): **n=10**

No. of efficient points	***range between*** $(\sum_{i \in N} c_i, T_{\max})$	***range between*** (T_{SPT}, T_{EDD})
4	23	6
2	30	4
7	23	7
5	44	5
2	25	3
1	43	0
10	52	10
4	33	7
10	27	12
6	41	20

Conclusions and suggestions

This paper considers the problem of scheduling families of jobs on a single machine to find all efficient solutions for $\sum_{i \in N} c_i$ and $T_{\max}$, where a set up time is in incurred whenever the machine switches from processing a job in one family to a job in another family.

An algorithm presents to solve the problem and find all efficient solutions for the case that the set up time δ_1 in SPT –rule is less that δ_2.For all the solved examples we see that there is a relation between number of efficient solutions and the range of (T_{SPT}, T_{EDD}), that is : No. of efficient solution $\leq$ range of $(T_{SPT}, T_{EDD}) + 1$.

The second case $\delta_1 > \delta_2$ in SPT –rule is not solvable by this algorithm, we gave an example that the algorithm is not work with this case.

Many ideas appear to solve scheduling problem with set up times and using efficient solutions, for example the range of tardiness $(T_{\max} - T_{\min})$ function.

Appendix A

The computer results for (n=20, 30, 40, 50)

.n=20

No. of efficient points between (T_{SPT}, T_{EDD})	***Range between*** $(\sum_{i \in N} c_i,\ T_{\max})$	***Range***
14	134	20
13	145	19
9	273	21
23	159	35
18	373	33
20	163	26
17	455	21
19	183	27
16	109	35
21	391	29

n=30

23	254	32
42	233	65
37	323	43
44	299	78
64	628	73
78	943	99
51	802	65
84	663	88
91	912	92
96	534	132

n=40

86	2354	96
113	1030	134
94	2003	139
82	1244	125
87	3425	103
54	8743	102
112	5112	145
97	7233	111
77	8827	82
82	7941	139

n=50

113	12943	234
242	10654	314
119	11423	233
211	91654	1032
104	43043	654
248	14343	1932
132	76234	7134
101	11488	543
234	54322	7831
112	83452	886

References

- Bischoff, M. and Klamroth, K. (2007): An Efficient Solution Method for Weber Problems With Barriers Based On Genetic Algorithms, European journal of operational research, Vol. 177, pp. 22-41.
- Crauwels, H. A. J., Potts, C. N., Van Oudheusden, D. and Van Wassenhove, L.V., (2005): Branch and Bound Algorithm for Single Machine Scheduling with Batching to Minimize The Number of Late Jobs, Journal of scheduling, Vol. 8, pp. 161-177.
- Lowe. J., Thisse, J. F., Ward, J. and Wendell, R. E. (1984): On Efficient Solutions to Multiple Objective Mathematical Programs, Management science, Vol. 30, pp.1346-1349.
- Potts, C. N. and Mikhail, Y. K., (2000): Scheduling with Batching, A Review, European Journal of Operational Research, Vol.120, pp.228-249.
- Rocha, P. L., Ravetti, M. G., Mateus, G. R. and Pardalos, P. M.(2008): Exact Algorithm For A Scheduling Problem With Unrelated Parallel Machines And Sequence And Machine-Dependent Setup Times, Computers and operations research, Vol. 35, pp. 1250-1264.
- Shabtay, D. and Steiner, G. (2007): Single Machine Batch Scheduling to Minimize Total Completion Time and Resource Consumption Costs, Journal of scheduling, Vol. 10, pp. 255-261.
- Steiner, S. and Radzik, T. (2008): Computing All Efficient Solutions Of The Biobjective Minimum Spanning Tree Problem, Computers and operations research, Vol. 35, pp.198-211.
- Van Wassenhove, L.N. and Gelders, L.F. (1980): Solving a Bicriterion Scheduling Problem, European journal of operational research, Vol. 4, pp. 42-48.
- Ward, J., (1989): Structure of Efficient Sets For Convex Objectives, Mathematics Of Operations Research, Vol. 14, pp.249-257.
- Yuan, J. J., Shang, W. P. and Feng, Q. (2005): A Note On The Scheduling With Two Families Of Jobs, Journal of scheduling, Vol. 8, pp.537-542.

خوارزمية معدلة لمسألة الجدولة مع الحلول الفعالة

أياد محمد رمضان
كلية العلوم ــ جامعة السليمانية

تاريخ الاستلام:٢٥/٥/٢٠٠٨، تاريخ القبول:٢/٤/٢٠٠٩

الخلاصة

من النتاجات لترتيب على ماكنة واحدة.عنوننا دالة مركبة حيث تتضمن تصغير مجموع الاتمام واكبر تأخير. (n) اعتبر ميزنا جميع الحلول الفعالة ثم الحل الامثل . قدمت خوارزمية معدلة لايجاد كل الحلول الفعالة للمسألةمع اوقات النصب. هذه الخوارزمية تتعامل مع حالة عندما تكون فيها اوقات النصب في ترتيب اقصر وقت اتمام (SPT) متزايدة .كما قدمت مثال مخالف للخوارزمية عندما تكون اوقات النصب في ترتيب اقصر وقت اتمام متناقص.

و مهمتنا هي تقديم صاحب القرار كل الحلول المحتملة للمسألة وتركه ليختار القرار النهائي.صاحب القرار له دالتين في اختياره ($\sum_{i \in N} c_i$) و $(T_{\max})$. وبعض الحلول (الفعالة) وسوف يختار احسن حل من بين الحلول ومعتمدا على خبراته.