Global Journal of Science Frontier Research
Volume 11 Issue 3 Version 1.0 May 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
ISSN: 0975-5896

A Model on Changeover Times with Single Machine Problem

By S. Mujeeb-Uddin, Kaleem A. Quraishi
Gandhi Faiz-e-Aam College, Shahjahanpur

Abstracts : The aim of this note is to discuss single machine problems with changeover times in connection with general shop problems.

GJSFR-F Classification : *FOR Code: 010399*

Strictly as per the compliance and regulations of:

A Model on Changeover Times with Single Machine Problem

S. Mujeeb-Uddin[α] And Kaleem A. Quraishi[Ω]

Abstract : The aim of this note is to discuss single machine problems with changeover times in connection with general shop problems.

I. Introduction

In this paper we consider scheduling problems in which the set I of all jobs or all operations (in connection with shop problems) in partitioned into disjoint sets $I_1, I_2, I_3, \cdots, I_r$ called groups, i.e. $I = I_1 \cup I_2 \cup I_3 \cup \cdots \cup I_r$ and $I_f \cap I_g = \phi$ for $f, g \in \{1,2,3,\cdots,r\}, f \neq g.$ Let N_j be the number of jobs in $I_j.$ Furthermore, we have the additional restrictions that for any two jobs (operations) i, j with $i \in I_f$ and $j \in I_g$ to be processed on the same machine M_k job (operation) j cannot be started until S_{fgk} time units after the finish time of job (operation) i or job (operation) $j.$ In a typical application, the groups correspond to different types of jobs (operations) and S_{fgk} may be interpreted as a machine dependent changeover time. During the changeover period, the machine cannot process another job. We assume that $S_{fgk} = 0$ for all $f, g \in \{1,2,3,\cdots,r\},\ k \in \{1,2,3,\cdots,m\}$ with $f = g$ and that the triangle inequality holds.

$$S_{fgk} + S_{ghk} \geq S_{fhk} \text{ for all } f, g, h \in \{1,2,3,\cdots,r\},\ k \in \{1,2,3,\cdots,m\}. \quad (1)$$

Both assumptions are realistic in practice.

The model can also handle set-up times for starting the first operations on the machines. For this purpose, we introduce a dummy operations belonging to a dummy group I_0 as an initial job, then the set-up time for an operation from group I_h starting on M_k can be represented by $S_{ohk}.$

If we consider single machine problems or if the changeover times are machine independent, we replace S_{fgk} by $S_{fg}.$ If the changeover time do not depend on both group I_f and $I_g,$ but only on the group I_g to which the job to be processed next belongs then we replace S_{fg} by $S_g.$ In the later case, the changeover times are called sequence independent, contrary to the general case in which they are called sequence dependent. If $S_{fg} = S$ for all $f, g = 1,2,3,\cdots,r$ then we have constant changeover times.

To indicate problems with changeover times, we add $\beta \in \{S_{fgk}, S_{fg}, S_g, S\}$ to the part of our general classification scheme.

1.1 *Single Machine Problems*

While problems $1|S_{fg}|C_{\max}$ and $1|S_g|L_{\max}$ are NP(Bruno and Downey (1978)), problem $1|S_g|C_{\max}$ can be solved polynomially by scheduling the jobs group by in any order.

About[α]: Department of Mathematics, Gandhi Faiz-e-Aam College, Shahjahanpur-242001, U.P., India.
E-mail: syedmujeebuddin.gfc@gmail.com
About[Ω]:Mathematics Section, Mewat Engineering College (Wakf),
Palla, Nuh, Mewat-122107, Haryana, India
E-mail: kaleemspn@yahoo.co.in

Next we will present dynamic programming procedures for the problems $1|S_{fg}L_{\max}|S_{fg}|\sum w_iC_i$ and $1|S_{fg}|\sum w_iC_i$ which are due to Monma and Potts [1989]. The following theorem is the basis for these procedures.

II. Main Result

Theorem 1:

a) For problem $1|S_{fg}|\sum w_iC_i$, there exists an optimal schedule where the early jobs within each group are ordered according to nondecreasing due dates.

b) For problem $1|S_{fg}|\sum w_iC_i$, there exists an optimal schedule where the jobs within each group are ordered according to nondecereasing $\frac{p_i}{w_i}$ values.

c) For problem $1|S_{fg}|L_{\max}$, there exists an optimal schedule where the jobs within each group are ordered according to nondecreasing due dates.

Proof :

Consider a schedule of the form D, j, E, i, F, where job i and j are from the same group and D, E, F, represent arbitrary blocks, i.e. partial sequences of jobs. Due to the triangle inequality (1), the total changeover times will not increase if D, j, E, i, F is replaced by D, i, j, E, F. We have to show that if $d_i < d_j$ for early jobs i, j or $\frac{p_i}{w_i} < \frac{p_j}{w_j}$ or $d_i < d_j$ then $\sum \omega_i U_i$ or $\sum \omega_i C_i$ or the $L_{\max}$ value will not increase when moving from the sequence D, j, E, i, F to one of the two sequence D, E, i, j, F or D, i, j, E, F.

a) If we have the objective function $\sum w_iU_i$ then the fact that i is early in D, j, E, i, F implies that i, j are early in D, E, i, j, F because $d_i < d_j$.

b) Consider the objective function $\sum w_iC_i$ and let $\frac{p_i}{w_i} < \frac{p_j}{w_j}$. It is convenient to replace changeover times by set-up jobs with processing time equal to the changeover time and zero weight. Furthermore, we assume that all set ups after j and before i and D, j, E, i, F are included in the partial sequence E. Define

$$p(E) = \sum_{i \in E} p_i \text{ and } w(E) = \sum_{i \in E} w_i$$

then an easy calculation shows that job j and block E can be swapped without increasing the objective value if $\frac{p_j}{w_j} > \frac{p(E)}{w(E)}$. Similarly, i and E can be swapped if $\frac{p(E)}{w(E)} > \frac{p_i}{w_i}$. In the first case we first swap j and E and then j and i without increasing the objective value. This provides the sequence D, E, i, j, F if $\frac{p_j}{w_j} \leq \frac{p(E)}{w(E)}$ then $\frac{p_i}{w_i} < \frac{p(E)}{w(E)}$ and we get D, i, j, E, F after two swaps.

c) Consider the objective function $L_{\max}$ and let $d_i < d_j$. Again, changeover times are replaced by set-up jobs with the changeover time and a very large due date. To see the effect of swapping a job j scheduled before a block $E = i_r, i_{r-1}, \cdots, i_1$ we replace E by a job with processing time $p_i, p_{r-1} = p(E)$ and a due date d_i, i_{r-1}, i_i where d_r, i_{r-1}, i_i is calculated by the recursion

$$d_{i\vartheta+l\cdots i_l} = \min\{d_{i\vartheta\cdots i_l}\ d_{i\vartheta+1} + \sum_{k=1}^{\vartheta} p_{ik}\} \text{ for } \vartheta = 1,2,3,\cdots,r-1$$

This follows by induction using the fact that, for two jobs i and j with finish times C_j and $C_i = C_j - p_j$, we have

$$\max\{C_j - d_j, C_j - p_j - d_i\} = C_j - \min\{d_j, d_i + p_j\}$$

Now we can proceed as in part (b). If $d_j > d_E := d_{ir-i_{r-1},1,\cdots i_1}$ then we can swap first j and E and then j and i.

For the following dynamic programming algorithms, we assume that the jobs each group are ordered according to theorem 1. We get derive an algorithm for problem $1|\ S_{fg}\sum w_i C_i$

We define $C(n_1, n_2, \cdots, n_r, t, h)$ to be the minimum cost of a partial schedule containing the first n_j jobs of groups $l_j\ (j = 1,2,\cdots,r),$ where the last job scheduled comes from group l_h and is completed at time $t,$ we have

$$0 \le n_j \le N_j \text{ for } j = 1,2,\cdots,r$$

and

$$0 \le t \le T := \sum_{i\in l} p_i + \sum_{j=1}^{r} p_j + N_j \max\{S_{f\,j}\ |1 \le f \le r\}$$

The recursion is

$$C(n_1, n_2, \cdots n_r, t, h) = \min\{C(n_1', n_2', \cdots, n_r', t', f) + w_{nh}^h\, t\ |1 \le f \le r\} \quad (2)$$

where $n_j' = n_j$ for $j \ne h, n_h' = n_n - 1,\ t' = t - p_{n_h}^h - s/h$ and $w_{n_h}^h$ and $p_{n_h}^h$ are the weight and processing time of the n_h-th job in groups l_h.

initially, we set $C(0,0,\cdots,0) = 0$ and all other $C-$values to infinity. The optimal schedule cost is found by selecting the smallest value of the form $C(N_1, N_r, t, h)$ for some schedule completion time $0 \le t \le T$ and some group I_h to which the final job belongs.

Because the number of states is bounded by $O(r\, n^r\, T)$ and (2) can be calculated in $O(r)$ steps, we have an $O(r^2\, n^r\, T)-$algorithm

Alternatively, we can replace the state variable t by variables $t_{fh}(f, h = 1,2,\cdots,r)$ representing the number of set-ups from group f to group $h.$ Note that t is readily computed from the state variables using.

$$t = \sum_{f,h=1}^{r} t_{fh} S_{fh} + \sum_{h=1}^{r}\sum_{\upsilon=1}^{nh} p_{\upsilon}^h$$

The complexity of this version is $O(r^2 n^{r+s})$, where s is the number of different values for changeover times. Note that $s \le r^2$ in general and $s \le r$ in the case of sequence-independent changeover times. If the number of groups is fixed we have a polynomial algorithm.

Similarly, we solve problem $1|S_{fg}|L_{\max}$. In this case (2) is replaced by

$$C(n_1, n_2, \cdots n_r, t, h) = \min\{\max\{C(n_1', n_2', \cdots, n_r', t', f),\ t - d_{nk}^h\} | 1 \le f \le r\} \quad (3)$$

The weighted number of late jobs problem is solved differently. By theorem 1(a), we only know an order for family jobs; the algorithm must also determine those jobs that are late. The late jobs may be appended in any order to the schedule of on-time jobs.

We define $C(n_1, n_2, \cdots n_r, t, h)$ to be the minimum weighted number of late jobs for the partial schedule containing the first n_j jobs of group l_j $(j = 1, 2, \cdots, r)$ where the last on-time job comes from group l_h and is completed at time t. We have

$$0 \le n_j \le N_j \text{ for } j = 1, 2, \cdots, r$$

and

$$0 \le t \le T := \min\{\max_{i \in l} d_i \sum_{i \in E} p_i + \sum_{j=1}^{r} N_j \max\{S_{fi} | 1 \le f \le r\}\}$$

The recursion is

$$C(n_1, n_2, \cdots n_r, t, h) = \begin{cases} \min\{\min_{1 \le f < r} C(n_1', n_2', \cdots, n_r', t', f) \\ C(n_1', n_2', \cdots, n_r', t, h) + w_{n_h}^h\} \text{ if } t \le d_{n_h}^h \\ C(n_1', n_2', \cdots, n_r', t, h) + w_{n_h}^h \text{ if } t \le d_{n_h}^h \end{cases}$$

where $n_j' = n_j$ for $j \ne h, n_h' = n_n - 1,\ t' = t - p_{n_h}^h - S_{fh}$ (4)

The initial values are

$$C(n_1, n_2, \cdots, n_r, 0, 0) = \sum_{j=1}^{r} \sum_{\upsilon=1}^{n_j} w_\upsilon^j$$

for $0 \le n_j \le N_j$ where $j = 1, 2, \cdots, r$. All other initial values are set to infinity.

The minimum weighted number of late jobs is found by selecting the smallest value of the form $C(N_1, N_2, \cdots, N_r, t, h)$ for some completion time of on time jobs, where $0 \le t \le T$ and for some group l_h containing the final on-time job.

The complexity is $O(r^2 n^r T)$. Again, it may be desirable to eliminate the state variable t from the recursion. The achieve this, we switch the state variable t with the objective function value as follows.

Define $C(n_1, n_2, \cdots n_r, w, h)$ to be the minimum completion time of on time jobs for a partial schedule containing the first n_j jobs of each group. l_j where the weighted number of late jobs is equal to w, and the last on-time job comes from group l_h. the initial values are $C(n_1, n_2, \cdots n_r, w, 0) = 0$, where $w = \sum_{j=1}^{r} \sum_{\upsilon=1}^{n_j} w_\upsilon^j$ for $0 \le n_j \le N_j (j = 1, 2, \cdots, r)$ and all other values are set to infinity.

The recursion is

$$C(n_1, n_2, \cdots n_r, w, h) = \min\{\min_{1 \le f \le r}\{C(n_1, n_2, \cdots n_r, w, f) + p' |$$

$$C(n_1, n_2, \cdots n_r, w, f) + p' \le d^h_{n_h}\}\ C(n_1', n_2', \cdots n_r', w', h)\}$$

where $n'_j = n_j$ for $j \neq h$, $n'_h = n_h - 1, p' = p^h_{n_h} + S_{fh}$ and $w' = w - w^h_{n_h}$

As in (4), the first term in the minimization chooses the $n_h -$ th job n group I_h to be scheduled on time, if possible, and chooses the previous on-time job from group I_f the second term selects the n_h th job of group I_h to be late.

The minimum weighted number of late jobs are found by selecting the smallest w for which $\min_{0 \le h \le r} C(N_1, N_2, \cdots, N_r, w, h)$ is finite.

The complexity is $O(r^2 n^r W)$, where $W = \sum_{i \in l} w_i$. It reduces to $O(r^2 n^{r+1})$ for the total number for late jobs problem.

References Références Referencias

1) Brucker, P. and Thiele, O.; A branch and bound method for the general shop problem with sequence dependent setup-times, *Osnabrucker Schriftenzur mathematic, Reihe P, Heft*, 168.
2) Bruno, J. L. and Downey, P.; Complexity of task sequencing with deadlines, setup times and changeover costs, SIAM J. comput., 7 (1978), 393-404.
3) Carlier, J.; One machine sequencing problem, *European J. operations research*, 11 (1982), 42-47.
4) Monma, C. L. and Potts, C. N.; On the complexity of scheduling with batch setup times, operations research, 37 (1989), 798-804.
5) Naddef, D. and Santos, C.; One pass batching algorithms for the one machine problem, *discrete applied mathematics*, 21 (1998), 133-145.
6) Panwalker, S., Smith, M. and Seidmann, A.; Common due date assignment to minimize total penalty for the one machine scheduling problem, *operations research*, 30 (1982) 391-399.