Minimizing Flow Time on a Single Machine with Job Families and Setup Times

Malapert Arnaud, Nattaf Margaux

Minimizing Flow Time on a Single Machine with Job Families and Setup Times.

Malapert Arnaud[1] and Nattaf Margaux[2]

[1] Université Côte d'Azur, CNRS, I3S, France
arnaud.malapert@univ-cotedazur.fr
[2] Univ. Grenoble Alpes, CNRS, Grenoble INP**, G-SCOP, 38000 Grenoble, France
margaux.nattaf@grenoble-inp.fr

Keywords: single machine scheduling ; flow time ; job families ; setup times

Context and problem description In (Mason and Anderson 1991), the authors consider a one machine scheduling problem with the goal of minimizing the average weighted flow time. In this problem, jobs are grouped into families with the property that a setup time is required only when processing switches from jobs of one family to jobs of another family. Furthermore, setup times are additive: each setup for a new family of job consists of the setdown from the previous family followed by a setup of the new family. These additive setups are considered as sequence-independent. Note that a setup is required at time 0 before the very first job is processed. For this problem, they define several properties of an optimal solution. These properties are then used in a branch-and-bound procedure.

If jobs belonging to the same family have equal processing times and unit weights, the application of their results directly leads to a polynomial time algorithm. However, if no setup time is required at time 0, i.e. before the very first job, their results cannot be applied directly. The first goal of this paper is to prove this affirmation. The second objective is to show how their results can be adapted in this special case. These results are then used to define a polynomial time algorithm to compute the optimal flow time for this special case. More generally, the objective is to use this algorithm for solving a parallel machine scheduling problem with time constraints and machine qualifications described in (Malapert and Nattaf 2019).

The problem considered in this paper is now formally described. Consider the problem of scheduling a set $\mathcal{N}$ of n jobs on one machine. Each job belongs to a family $f \in \mathcal{F}$ and the family associated with a job j is denoted by f_j. Each family is then associated with a number of jobs to process n_f, a processing time p_f, and a sequence independent setup time s_f. Thus, switching the production from family f to $f' \neq f$ will require a setup time $s_{f'}$. Note that no setup time is needed between two jobs of the same family. The important difference to the original problem is that the setup times are not additive anymore because no setup time is required at time 0 before the very first job is processed. The goal is to minimize the flow time which is the sum of the finishing time of all jobs.

Optimal solution properties The goal of this section is to describe several properties and characteristics of an optimal solution. The properties will then be used to design our polynomial-time algorithm.

To represent a solution, let S be a sequence representing an ordered set of n jobs. Then, S can be seen as a series of blocks, where a block is a maximal consecutive subsequence of jobs in S from the same family. Let B_i be the i-th block of the sequence $S = \{B_1, B_2, \ldots, B_r\}$.

** Institute of Engineering Univ. Grenoble Alpes

Successive blocks contain jobs from different families. Therefore, there will be a setup time before each block (except the first one). The idea is to adapt SPT Smith's rule (Smith 1956) for blocks instead of individual jobs. To this end, blocks are considered as individual jobs with processing time P_i and weight W_i with: $P_i = s_{f_i} + |B_i| \cdot p_{f_i}$ and $W_i = |B_i|$ where f_i denotes the family of jobs in B_i (which is the same for all jobs in B_i).

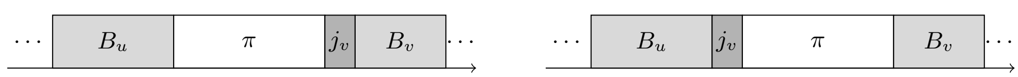
The following theorem states that there exists an optimal solution S containing exactly $|\mathcal{F}|$ blocks and that each block B_i contains all jobs of the family f_i.

Theorem 1. *Let $\mathcal{I}$ be an instance of the problem. There exists an optimal solution $S^* = \{B_1, \ldots, B_{|\mathcal{F}|}\}$ such that $|B_i| = n_{f_i}$ where f_i is the family of jobs in B_i.*

Proof. Consider an optimal solution $S = \{B_1, \ldots, B_u, \ldots, B_v, \ldots, B_r\}$ with two blocks B_u and B_v $(u < v)$, containing jobs of the same family $f_u = f_v = f$. We show that moving the first job of B_v at the end of block B_u can only improve the solution.

Let us define P and W as: $P = \sum_{i=u+1}^{v-1} P_i + s_f$ and $W = \sum_{i=u+1}^{v-1} |B_i|$. Note that P (resp. W) is the total time of all jobs, including setups, (resp. the number of jobs) performed between the last job of B_u and the first job of B_v. Let us call π this partial sequence.

Let S' be the sequence formed by moving the first job of B_v, say job j_v, at the end of block B_u , that is swapping the position of π and j_v (see Fig. 1).



(a) Sequence S (b) Sequence S'

Fig. 1. Construction of sequence S' from sequence S.

First, the difference on the flow time is computed in the cases where $|B_v| > 1$ and where $|B_v| = 1$.
If $|B_v| > 1$ (that is there are still jobs left in B_v after the removal of job j_v), then the flow times of jobs sequenced before B_u or after B_v do not change. All the jobs in π have their flow times increased by p_f and j_v has its flow time decreased by P, giving a difference in the total flow time of:

$$FT_{S'} - FT_S = W \cdot p_f - P$$

If $|B_v| = 1$, then B_v is left with no jobs after moving job j_v, and so the setup time associated with B_v is deleted from the sequence. This reduces the flow times of all jobs sequenced after π by the amount of s_f, giving an additional reduction in the total net flow time:

$$FT_{S'} - FT_S = W \cdot p_f - P - \sum_{i=v+1}^{r} |B_i| \cdot s_f$$

Hence, if one can prove that $P/W \geq p_f$, $FT_S - FT_{S'} < 0$ and the flow time can only be improved in S'.

Lemma 1. $\frac{P}{W} \geq p_f$

Proof. Consider the sequence S and suppose that $p_f > P/W$. Let S'' be the sequence formed by moving the last job of B_u at the beginning of block B_v. If $|B_u| > 1$, then the difference of flow time is: $FT_{S''} - FT_S = P - W \cdot p_f$. Since, by definition, $p_f > P/W$, we have that $FT_{S''} - FT_S < 0$ which contradict the fact that S is optimal.

If $|B_u| = 1$, $FT_{S''} - FT_S = \begin{cases} W \cdot p_f - P - \sum_{i=v+1}^{r} |B_i| \cdot s_f & \text{if } u \neq 1 \\ W \cdot p_f - P - \sum_{i=v+1}^{r} |B_i| \cdot s_f - \sum_{i=u+1}^{r} |B_i| \cdot s_{f_{u+1}} & \text{if } u = 1 \end{cases}$
And then, S'' is a better solution than S which is a contradiction. Hence, we have $P/W \geq p_f$. □

Therefore, by Lemma 1, $FT_{S'} - FT_S \leq 0$. Hence, swapping the position of job j_v with π leads to a solution S' at least as good as S. Repeated applications of this operation yield the result.

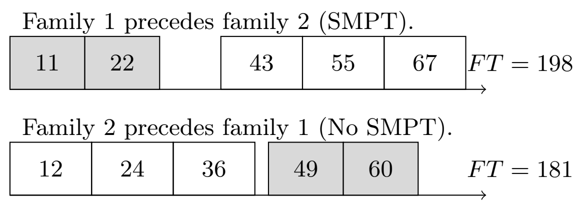
The mean processing time of a block B_i can be defined as $MPT(B_i) = P_i/W_i$. Theorem 2 generalized the SPT rule for blocks.

Theorem 2. *In an optimal sequence of the problem, the blocks 2 to $|\mathcal{F}|$ are ordered by SMPT (Shortest Mean Processing Time). That is, if $1 < i < j$ then $MPT(B_i) \leq MPT(B_j)$.*

This theorem is not exactly the same as the one in (Mason and Anderson 1991). Indeed, their theorem allows the ordering of all blocks according the SMPT rule while Theorem 2 orders only blocks 2 to $|\mathcal{F}|$. Figure 2 gives a counterexample showing that the SMPT rule is not optimal when there is no setup at time 0.

f	(n_f	$\times\ p_f$	$+\ s_f$)	$\div\ n_f = MPT_f$
1	2	11	2	12
2	3	12	9	15

(a) Problem instance with 2 families and 5 jobs.



(b) Scheduling jobs of a family before those of the other (Numbers are completion times of the jobs).

Fig. 2. The SMPT rule may lead to suboptimal solutions when no setup time is required at time 0.

Proof (Th. 2). Let $S = \{B_1, B_2, \ldots, B_u, B_{u+1}, \ldots, B_{|\mathcal{F}|}\}$ be a sequence in which blocks B_u and B_{u+1}, $1 < u \leq u+1 \leq |\mathcal{F}|$, are not in $SMPT$ order; that is, $MPT(B_u) > MPT(B_{u+1})$. Consider the change in total flow time if the processing order of B_u and B_{u+1} is reversed. Clearly the flow times of any jobs originally scheduled before run B_u will not be changed. Also, the total time to complete blocks B_u and B_{u+1} will not be changed, so the flow times of any jobs scheduled after B_{u+1} will not be altered. Hence, only the flow times of the jobs in blocks B_u and B_{u+1} needs to be considered. Each job in B_{u+1} has its completion time reduced by P_u and each job in B_u has its completion time increased by P_{u+1}. Thus, the change in weighted flow time is given by:

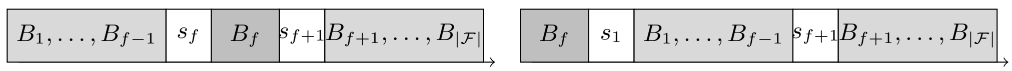
$$\Delta F_w = W_u \cdot P_{u+1} - W_{u+1} \cdot P_u$$

But, since $MPT(B_u) > MPT(B_{u+1})$, $P_u \cdot W_{u+1} > P_{u+1} \cdot W_u$, and so $\Delta F_w < 0$.

The following section explains how these results are used to define a polynomial time algorithm for solving the problem.

Polynomial-time algorithm Theorem 1 states that there exists an optimal solution S containing exactly $|\mathcal{F}|$ blocks and that each block B_i contains all jobs of family f_i. Theorem 2 states that the blocks B_2 to $B_{|\mathcal{F}|}$ are ordered by $SMPT$. Finally, one only needs to determine which family is processed in the very first block.

Algorithm 1 take as input the jobs grouped in blocks and in SMPT orders. The algorithm starts by computing the flow time of this schedule. Each block is then successively moved to the first position (see Figure 3) and the new flow time is computed. The solution returned by the algorithm is therefore the one achieving the best flow time.



(a) SMPT Sequence (b) B_f is moved in the first position.

Fig. 3. SMPT Sequence and Move Operation.

For sake of readability, let $F(f)$ denote the internal flow time of the block f, *i.e* the flow time of its jobs when starting at time 0: $\sum_1^{n_f} i \times p_f$

Algorithm 1: SMPT Scheduling without setup at time 0.

Data: n_f, p_f, s_f for $f \in \mathcal{F}$ in SMPT order ($MPT_f \leq MPT_{f+1}$) such that $n_f > 0$.

Result: The optimal flow time FT.

`// Compute the flow time` FT_1 `of the SMPT sequence` $\{B_1, \ldots, B_{|\mathcal{F}|}\}$

$FT_1 \leftarrow F(1)$; $P \leftarrow n_1 \times p_1$;

for $f \leftarrow 2$ **to** $|\mathcal{F}|$ **do**

  `// shift jobs of` B_f `and add their internal flow times`

  $FT_1 \leftarrow FT_1 + n_f \times (P + s_f) + F(f)$

  $P \leftarrow P + s_f + (n_f \times p_f)$ `// Ending time of` B_f

`// Note that` P `is the makespan of the SMPT sequence`

`// Compute the flow time when the block` B_f `is in the first position`

$FT \leftarrow FT_1$; $W \leftarrow n$;

for $f \leftarrow |\mathcal{F}|$ **to** 2 **do**

  $W \leftarrow W - n_f$ `// Number of jobs in` $\{B_1, \ldots, B_{f-1}\}$

  $P \leftarrow P - s_f - (n_f \times p_f)$ `// Ending time of` B_{f-1} `in the SMPT sequence`

  `// Compute the variations of the flow time`

  $\Delta_f \leftarrow -(P + s_f) \times n_f$ `// For` B_f

  $\Delta^- \leftarrow (n_f \times p_f + s_1) \times W$ `// For` $\{B_1, \ldots, B_{f-1}\}$

  $\Delta^+ \leftarrow (s_1 - s_f) \times (n - n_f - W)$ `// For` $\{B_{f+1}, \ldots, B_{|\mathcal{F}|}\}$

  $FT_f \leftarrow FT_1 + \Delta^- + \Delta_f + \Delta^+$ `// For the entire sequence`

  if $FT_f < FT$ **then** $FT \leftarrow FT_f$ `// Update the best flow time`

The complexity for ordering the families in SMPT order is $O(|\mathcal{F}| \log |\mathcal{F}|)$. The complexity of Algorithm 1 is $O(|\mathcal{F}|)$. So, the complexity for finding the optimal flow time is $O(|\mathcal{F}| \log |\mathcal{F}|)$.

References

Malapert A., Nattaf M., 2019, “A New CP-Approach for a Parallel Machine Scheduling Problem with Time Constraints on Machine Qualifications” *CPAIOR 2019*, pp. 426-442.

Mason, A. J. and Anderson, E. J., 1991, “Minimizing flow time on a single machine with job classes and setup times” *Naval Research Logistics (NRL)*, Vol. 38(3), pp. 333-350.

Smith, W. E., 1956, “Various optimizers for single-stage production” *Naval Research Logistics (NRL)*, Vol. 3(1-2), pp. 59-66.