

THE COMPLEXITY OF SOME FLOW-SHOP SCHEDULES WITH POSITIVE TASK-TIMES

VU DINH HOA

Abstract. The general flow-shop problem is known to be NP-complete. Solution have also been specified in several special cases. A j -maximal (j -minimal) flow-shop is a particular kind of flow-shop in which the j -th task of any job has the longest (shortest) execution time comparing to another tasks of this job. We prove in this paper that the problem to find an optimal schedule for three-stage j -maximal (i -minimal) ($i \neq 2$) flow-shop with positive task time is NP-complete.

Tóm tắt. Bài toán lịch biểu tổng quát vẫn được biết là bài toán NPC. Người ta xét và giải bài toán này trong nhiều lớp đặc biệt khác nhau. Một bài toán lịch biểu j -maximal (j -minimal) là bài toán lịch biểu đặc biệt khi thời gian gia công ở công đoạn thứ j là lớn nhất (hoặc nhỏ nhất) so với thời gian gia công ở các công đoạn khác đối với công việc đang tiến hành. Ta chứng minh trong bài này là vấn đề tìm lịch biểu tối ưu cho bài toán j -maximal (i -minimal) với 3 công đoạn ($i \neq 2$) với thời gian gia công mỗi công đoạn là dương, vẫn là NPC.

1. INTRODUCTION

Flow-shop [5] consists of $m \geq 1$ processors $\langle P_1, P_2, \dots, P_m \rangle$ and $n \geq 1$ jobs $\{J_1, J_2, \dots, J_n\}$. Each processor P_j performs a different task and each job J_i has a chain of m tasks. With T_{ji} we denote the j -th task of J_i on processor P_j with execution time t_{ji} . Flow-shop with positive task time is one with $t_{ji} > 0$ for all j and i . Furthermore, each task T_{ji} has to be processed on P_j and can only be executed after $T_{j-1,i}$ has been finished. A schedule for a flowshop is defined as a sequence of tasks to be executed by each processor. A schedule is called a *permutation* schedule if the schedule on each processor is the same. If we allow a task to be partitioned and done in several time intervals, the schedule is called preemptive. In the following we only consider nonpreemptive schedules for which a processor cannot be interrupted in between once it has begun executive of one task. Moreover, we denote the schedule length or finish time of a schedule φ is by $f(\varphi)$.

2. PROBLEM

OFT schedule (optimal finish time schedule) is one which has shortest finish time among all schedules. We can state the OFT-problems, problems to find an OFT schedule, as a language decision problem as follows:

FOFT-Problem. Given an m -processor n -job flow-shop and a number T , does there exist a schedule with length less than or equal to T ?

Johnson (see [4]) showed that the OFT-problem for two processors can be solved in $O(n \log n)$ time and suggested an algorithm for three stages case which only works in certain circumstances. However, the general FOFT-problem is known to be NP-complete (see [8]). Solution for the general OFT-problem have been specified for several other special cases. A j -maximal (j -minimal) flow-shop is a particular kind of flow-shop in which the j -th task of any job has the longest (shortest) execution time comparing to another tasks of this job.

Chin and Tsai [6] proved that the 2-minimal FOFT-problem remains a NP-complete, even for

the three-stage case, i. e. for the case $m = 3$. On the otherwise, Burn and Rooker [4] shown that Jonhson's polynomial algorithm works for the three stages 2-minimal flow-shop with positive task time.

Let L stand for the processor with the largest task of each job, S the processor with the smallest task and M for the remaining processor. Then three-stage flow-shop scheduling of type j -maximal and at the same time i -minimal ($i \neq j$) fall into six cases: LMS , LSM , MLS , MSL , SML and SLM . As we know, Burn and Rooker proved that Johnson's polynomial algorithm works for the 2-minimal three-stage flow-shop with positive task-times. Recently, Achugbue and Chin gave an algorithm with polynomial time for the cases LMS and SML for flow-shop with positive task time.

In the following we will show that the remaining cases MLS and SLM are NP-complete.

3. RESULTS AND PROOFS

First, note that FOFT-problem is in NP (see [11]) and PAR (see [6]) is a NP-complete problem and 3PAR (see [8]) is a strongly NP-complete problem.

PAR-problem. Given a multiset $S = \{a_1, a_2, \dots, a_n\}$ of nonnegative integers a_i with $\sum_{i=1}^n a_i = K$, does there exist a subset U of $\{1, 2, \dots, n\}$ such that $\sum_{i \in U} a_i = \frac{K}{2}$.

3PAR-problem. Given a multiset $S = \{a_1, a_2, \dots, a_{3n}\}$ of nonnegative integers with $\sum_{i=1}^{3n} a_i = nK$ such that $\frac{K}{4} < a_i < \frac{K}{2}$, does there exist a partition of S into n disjoint three subsets of integers such that each has a sum exactly equal to K .

Lemma 1. (Lemma 1 in [1]) The three-stage flow-shop $n + 2$ jobs:

$$t_{1i} = 2(i - 1)K, \quad t_{2i} = (2i - 1)K, \quad t_{3i} = 2(i + 1)K, \quad \text{for } 1 \leq i \leq n + 1,$$

$$t_{1,n+2} = t_{3,n+2} = 0, \quad t_{2,n+2} = t_{3,n+1},$$

has the unique optimal permutation schedule $(1, 2, \dots, n + 2)$ of finish time $(n^2 + 5n + 5)K$.

Lemma 2. (Lemma 2 in [1]) The three-stage flow-shop $n + 1$ jobs:

$$t_{1i} = (i^2 + 3i + 4) \frac{K}{2}, \quad t_{2i} = (i^2 + i + 4) \frac{K}{2}, \quad t_{3i} = (i^2 - i + 2) \frac{K}{2}, \quad \forall 1 \leq i \leq n + 1,$$

has the unique optimal permutation schedule $(n + 1, n, n - 1, \dots, 2, 1)$ with the finish time $f(\varphi) = \left(\sum_{i=1}^{n+1} \frac{(i^2 + 3i + 4)}{2} + 4 + n\right)K$.

Lemma 3. [4] An OFT-schedule for three-stage flow-shop with positive task time may be found among the permutation schedules.

In the following we will show that the j -maximal (i -minimal) ($i \neq j$) flow-shop with positive task time is NP-complete and, specially, that the remaining cases MLS and SLM are NP-complete.

Theorem 1. The FOFT-problem for three-stage 2-maximal flow-shop with positive task time is NP-complete.

Proof. From the multiset $S = \{a_1, a_2, \dots, a_n\}$ we construct the following three-stage 2-maximal flow-shop with positive task time and with $n + 1$ jobs.

$$t_{1,i} = \frac{K}{4n}, \quad t_{2,i} = a_i + \frac{K}{4n}, \quad t_{3,i} = \frac{K}{4n}, \quad \text{for } 1 \leq i \leq n,$$

$$t_{1,n+1} = \frac{K}{2} + \frac{K}{4n}, \quad t_{2,n+1} = \frac{K}{2} + \frac{(n-2)}{4n}K, \quad t_{3,n+1} = \frac{K}{2} + \frac{K}{4n},$$

where $\sum_{i=1}^n a_i =: K$ and $T = 2K$.

Now we will show that the FOFT-problem for the above flow-shop has a schedule with finish time $\leq 2K$ iff S has a partition U with $\sum_{i \in U} a_i = \frac{K}{2}$.

(a) If S has a partition U with $\sum_{i \in U} a_i = \frac{K}{2}$ then there is a schedule φ with finish time $2K$.

One of such schedule φ is shown in figure 1. Since

$$\sum_U t_{1,i} + t_{1,n+1} = \frac{K}{4n} + \sum_U t_{2,i},$$

the $n+1$ -th job begins immediately to be processed on the next processor after his task on a processor has been finished. Thus, the finish time of this schedule is given by the sum:

$$\begin{aligned} f(\varphi) &= \sum_{i \in U} t_{1,i} + t_{1,n+1} + t_{2,n+1} + t_{3,n+1} + \sum_{i \notin U} t_{3,i} \\ &= |U| \frac{K}{4n} + \left(\frac{K}{2} + \frac{K}{4n}\right) + \left(\frac{K}{2} + (n-2) \frac{K}{4n}\right) + \left(\frac{K}{2} + \frac{K}{4n}\right) + (n - |U|) \frac{K}{4n} \\ &= 2K. \end{aligned}$$

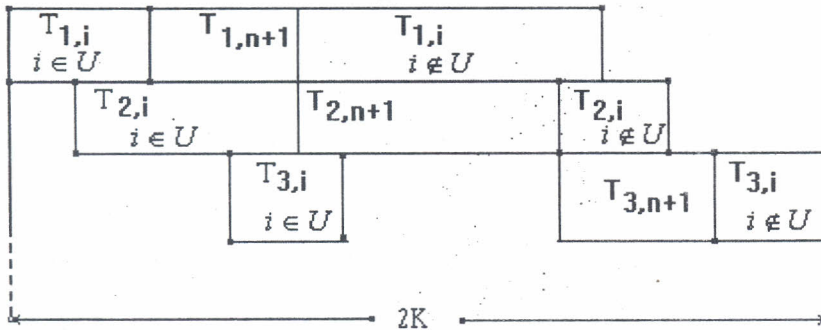


Figure 1

(b) If φ is schedule for our flow-shop with $f(\varphi) \leq 2K$, then S has a partition.

By Lemma 3, we can suppose that φ is a permutation schedule. We set

$$U_1 := \{i : \text{task } T_{1,i} \text{ finish before task } T_{1,n+1}\},$$

$$U_2 := \{i : \text{task } T_{1,i} \text{ finish after task } T_{1,n+1}\}.$$

For the case $U_1 \neq \emptyset$ we have:

$$\begin{aligned} 2K &\geq \frac{K}{4n} + \sum_{i \in U_1} t_{2,i} + t_{2,n+1} + t_{3,n+1} + \sum_{i \in U_2} t_{3,i} \\ &\geq \frac{3K}{2} + \sum_{i \in U_1} a_i. \end{aligned}$$

And therefore $\frac{K}{2} \geq \sum_{i \in U_1} a_i$ (also true for $U_1 = \emptyset$).

Similarly, for case $U_2 \neq \emptyset$:

$$\begin{aligned}
 2K &\geq \sum_{i \in U_1} t_{1,i} + t_{1,n+1} + t_{2,n+1} + \sum_{i \in U_2} t_{2,i} + \frac{K}{4n} \\
 &\geq \frac{3K}{2} + \sum_{i \in U_2} a_i.
 \end{aligned}$$

And therefore $\frac{K}{2} \geq \sum_{i \in U_2} a_i$ (also true for $U_2 = \emptyset$).

Since $U_1 \cup U_2 = \{1, 2, \dots, n\}$, we have $\frac{K}{2} = \sum_{i \in U_1} a_i = \sum_{i \in U_2} a_i$. Thus S has a partition U with $\sum_{i \in U} a_i = \frac{K}{2}$.

Corollary 1. *The FOFT-problem for the three-stage MLS and SLM flow-shop with positive task time is NP-complete.*

Theorem 2. *The FOFT-problem for three-stage 1-minimal flow-shop with positive task time is strongly NP-complete.*

Proof. Given an instance of 3PAR-problem with $S = \{a_1, a_2, \dots, a_{3n}\}$ of $3n$ nonnegative integers a_i such that $\sum_{i=1}^{3n} a_i = nK$ and $\frac{K}{4} < a_i < \frac{K}{2}$, we can construct the following 1-minimal flow-shop with $4n + 2$ jobs:

$$t_{1,i} = 2(i - 1)K + 1, \quad t_{2,i} = (2i - 1)K + 1, \quad t_{3,i} = 2(i + 1)K + 1, \quad \text{for } 1 \leq i \leq n,$$

and

$$t_{1,n+2} = 1, \quad t_{2,n+2} = t_{3,n+1}, \quad t_{3,n+2} = 1,$$

$$t_{1,i} = 1, \quad t_{2,i} = a_{i-n-2} + 1, \quad t_{3,i} = 1, \quad \text{for } n + 3 \leq i \leq 4n + 2,$$

and $T = (4n + 4) + (n^2 + 5n + 5)K$.

(a) If S has a 3-partition $\{U_1, U_2, \dots, U_n\}$ such that

$$\sum_{U_1} a_i = \sum_{U_2} a_i = \dots = \sum_{U_n} a_i = K$$

then the schedule showing in figure 2 has the finish time $T = (4n + 4) + (n^2 + 5n + 5)K$.

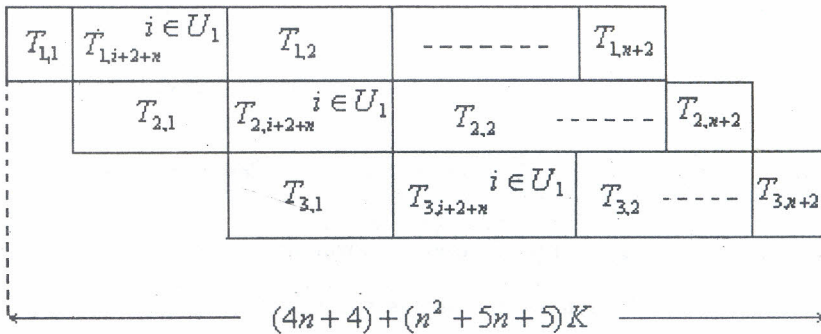


Figure 2

(b) If there is a schedule φ with finish time

$$f(\varphi) \leq (4n + 4) + (n^2 + 5n + 5)K.$$

By reducing each task of job exactly 1 unit time, φ is a schedule with finish time $(n^2 + 5n + 5)K$ for the following three-stage flow-shop \mathcal{F} with $4n + 2$ jobs:

$$t_{1,i} = 2(i - 1)K, \quad t_{2,i} = (2i - 1)K, \quad t_{3,i} = 2(i + 1)K, \quad \text{for } 1 \leq i \leq n,$$

and

$$t_{1,n+2} = 0, \quad t_{2,n+2} = t_{3,n+1}, \quad t_{3,n+2} = 0, \\ t_{1,i} = 0, \quad t_{2,i} = a_{i-n-2}, \quad t_{3,i} = 0, \quad \text{for } n + 3 \leq i \leq 4n + 2.$$

Without the last $3n$ jobs the three-stage flow-shop \mathcal{F}' with the first $n + 2$ jobs:

$$t_{1,i} = 2(i - 1)K, \quad t_{2,i} = (2i - 1)K, \quad t_{3,i} = 2(i + 1)K, \quad \text{for } 1 \leq i \leq n,$$

and

$$t_{1,n+2} = 0, \quad t_{2,n+2} = t_{3,n+1}, \quad t_{3,n+2} = 0.$$

has the unique permutation schedule $(1, 2, \dots, n + 2)$ with the same finish time $(n^2 + 5n + 5)K$ because of Lemma 1. Thus, the schedule φ is only an "extended" schedule of $(1, 2, \dots, n + 2)$, it means that the order of $(1, 2, \dots, n + 2)$ in φ remains the same and that by \mathcal{F} the three processors perform the last $3n$ jobs in the pause time of \mathcal{F}' . The only pause time by the schedule $(1, 2, \dots, n + 2)$ of \mathcal{F}' is established by the second processor and has the form of exactly n intervals with the same volume K (see Fig. 3). Since in \mathcal{F} we have $t_{2,i} = a_{i-n-2}, \forall i = n + 3, \dots, 4n + 2$, S has a partition into n subset U_1, U_2, \dots, U_n such that $\sum_{U_i} a_i = K$. Since $\frac{K}{4} < a_i < \frac{K}{2}$, each U_i contains exact 3 elements of S .

Thus S has a 3-partition.

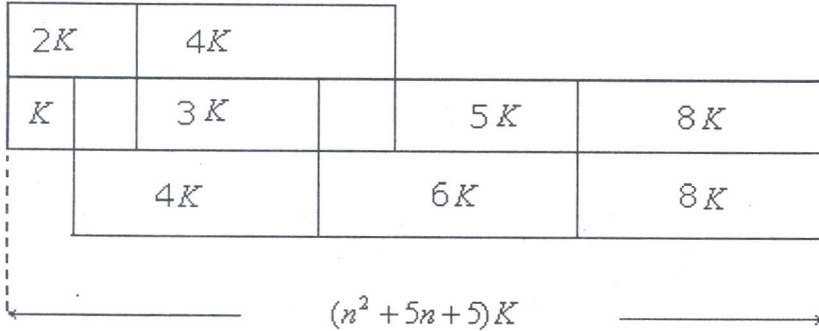


Figure 3 (One example with $n = 2$)

With similar proof to proof of Theorem 2 (by the symmetry of the first and the third processor) we contain the following corollary.

Corollary 2. *The FOFT-problem for three-stage 3-minimal flow-shop with positive task time is strongly NP-complete.*

Theorem 3. *The FOFT-problem for three-stage 1-maximal flow-shop with positive task time is strongly NP-complete.*

Proof. The proof is similar to the proof of Theorem 2. From an instance of 3-partition problem with the set $S = \{a_1, a_2, \dots, a_{3n}\}$ of $3n$ nonnegative integers a_i such that $\sum_{i=1}^{3n} a_i = nK$ and $\frac{K}{4} < a_i < \frac{K}{2}$, we can construct the following 1-maximal flow-shop with $4n + 1$ jobs:

$$t_{1,i} = (i^2 + 3i + 4)\frac{K}{2} + 1, \quad t_{2,i} = (i^2 + i + 4)\frac{K}{2} + 1, \quad t_{3,i} = (i^2 - i + 2)\frac{K}{2} + 1, \quad \text{for } 1 \leq i \leq n + 1,$$

and

$$t_{1,i} = t_{3,i} = a_{i-n-1} + 1, \quad t_{2,i} = 1, \quad \text{for } n + 2 \leq i \leq 4n + 1.$$

We will show that S contains an 3-partition iff there is a schedule φ with finish time $f(\varphi) \leq \left(\sum_{i=1}^{n+1} \frac{i^2 + 3i + 4}{2} + 4 + n\right)K + 4n + 3$.

(a) If S has an 3-partition $\{U_1, U_2, \dots, U_n\}$ then the permutation schedule $(n + 1, i \in U_1, n, i \in U_2, \dots, 2, i \in U_n, 1)$ (Fig. 4) has the finish time

$$f(\varphi) = \left(\sum_{i=1}^{n+1} \frac{i^2 + 3i + 4}{2} + 4 + n\right)K + 3n.$$

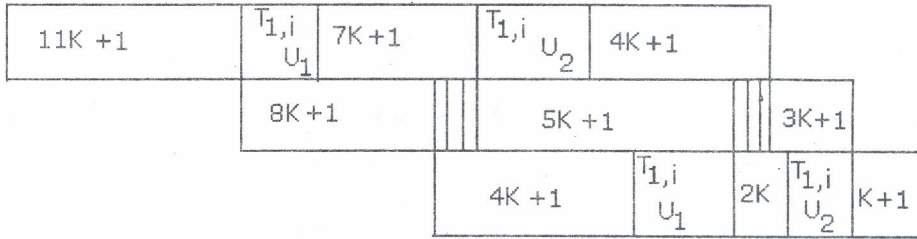


Figure 4 (One example with $n = 2$)

(b) If there is a schedule φ with finish time

$$f(\varphi) \leq \left(\sum_{i=1}^{n+1} \frac{i^2 + 3i + 4}{2} + 4 + n\right)K + 3n.$$

By reducing each task exactly 1 unit time, φ is a schedule with finish time $f(\varphi) \leq \left(\sum_{i=1}^{n+1} \frac{i^2 + 3i + 4}{2} + 4 + n\right)K$ for the following three-stage flow-shop \mathcal{F} with $4n + 1$ jobs:

$$t_{1,i} = (i^2 + 3i + 4) \frac{K}{2}, \quad t_{2,i} = (i^2 + i + 4) \frac{K}{2}, \quad t_{3,i} = (i^2 - i + 2) \frac{K}{2}, \quad \text{for } 1 \leq i \leq n + 1,$$

and

$$t_{1,i} = t_{3,i} = a_{i-n-1}, \quad t_{2,i} = 0, \quad \text{for } n + 2 \leq i \leq 4n + 1.$$

Without the last $3n$ jobs the three-stage flow-shop \mathcal{F}' with the first $n + 1$ jobs:

$$t_{1,i} = 2(i - 1)K, \quad t_{2,i} = (2i - 1)K, \quad t_{3,i} = 2(i + 1)K, \quad \text{for } 1 \leq i \leq n,$$

and

$$t_{1,n+2} = 0, \quad t_{2,n+2} = t_{3,n+1}, \quad t_{3,n+2} = 0.$$

has the unique permutation schedule $(n + 1, n, \dots, 1)$ with the same finish time $\left(\sum_{i=1}^{n+1} \frac{i^2 + 3i + 4}{2} + 4 + n\right)K$ because of Lemma 1. Thus, the schedule φ is only an "extended" schedule of $(n + 1, n, \dots, 1)$, it means that the order of $(n + 1, n, \dots, 1)$ in φ remains the same and that by \mathcal{F} the three processors perform the last $3n$ jobs in the pause time of \mathcal{F}' . The only pause time by the schedule $(n + 1, n, \dots, 1)$ of \mathcal{F}' is established by the third processor and has the form of exactly n intervals with the same volume K (see Fig. 5). Since in \mathcal{F} we have $t_{3,i} = a_{i-n-1}, \forall i = n + 2, \dots, 4n + 1$, S has a partition into n subset U_1, U_2, \dots, U_n such that $\sum_{U_i} a_i = K$. Since $\frac{K}{4} < a_i < \frac{K}{2}$, each U_i contains exact 3 elements of S . Thus S has a 3-partition.

With similar proof to proof of Theorem 5 (by the symmetry of the first and the third processor) we contain the following corollary.

Corollary 3. *The FOFT-problem for three-stage 3-maximal flow-shop with positive task time is strongly NP-complete.*

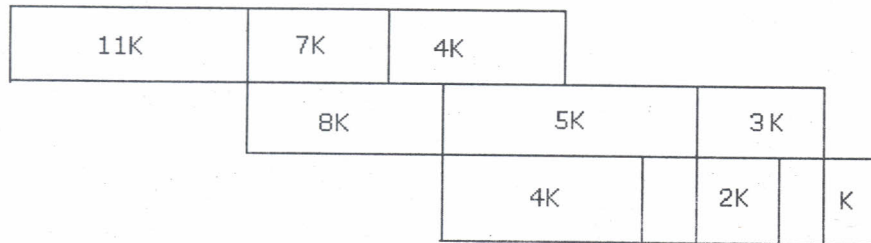


Figure 5 (One example with $n = 2$)

REFERENCES

1. Achugbue J. O. and Chin F. Y., Complexity and solution of some three-stage flowshop scheduling problems, *Math. Operat. Res.* **7** (4) (1982) 532-544.
2. Achugbue J. O., "The complexity of some deterministic scheduling problems", Ph. D. Thesis, Department of Computing Science University of Alberta, Edmonton, Spring, 1980,
3. Arthenari T. S. and Mukhopadhyay, A Note on a paper by W. Szwarc, *Naval. Mes.*, 1971.
4. Burns F. and Rooker J., Three-stage Flow-shop with Recessive Second Stage, *Oper. Res.* **26** (1978) 207-208.
5. Conway R. W. and Maxwell W. L., and Miller L. W., *Theory of Scheduling*, Addison-Wesley Reading Mass, 1967.
6. Chin F. Y. and Tsai L. L., On J -maximal and J -minimal Flowshop Schedules, *J. ACM* **28** (3) (1981).
7. Johnson S. M., Optimal two and three stage production schedules with setup times included, *Naval. Res. Logist. Quar* **1** (1954) 61-68.
8. Garey M. R., Johnson D. S., and Sethi R., The complexity of flowshop and jobshop scheduling, *Math. Oper. Res.* (1976) 117-129.
9. Smith M. L., Panwalleer S. S., and Duclek B. A., Flowshop sequencing problem with ordered processing time matrices, 1975.
10. Szwarc W., Optimal two-machine orderings in the $3 \times n$ flowshop problem, *Oper. Res.* **25** (1977) 70-77.
11. Ullman J. D., Complexity of scheduling problems, In: *Computer and Job/Shop Scheduling Theory*, E. G. Coffman Jr. (Ed.) Willey, 1976, 130-164.
12. Vu Dinh Hoa, Note on Flow-shop schedules with positive Task-times, Preprint no. 7, Institute of Computer Science and Cybernetics, Hanoi, 1987.

Received June 6, 1999