Information Processing Letters 116 (2016) 65–69

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Complexity of problem $TF2|v=1, c=2|C_{\max}$

Yan Lan [a,b], Xin Han [a,*,1], Zongtao Wu [a], He Guo [a], Xin Chen [a]

[a] Software School, Dalian University of Technology, Dalian 116620, China
[b] Dalian Neusoft Institute of Information, China

ARTICLE INFO

Article history:
Received 10 May 2014
Received in revised form 14 July 2015
Accepted 4 September 2015
Available online 10 September 2015
Communicated by S.M. Yiu

Keywords:
Analysis of algorithms
Flow shop
NP-hardness
Optimization

ABSTRACT

In this paper, we study a flow shop scheduling problem with an interstage transporter, in which there are a set of n jobs, two machines A and B, and one transporter located at machine A initially. Each job has to be processed on machine A first, then transported by a vehicle to machine B, finally processed on machine B. The interstage transporter can carry at most two jobs between the machines at a time. When the transporter carries items from machine A to machine B, it will take time τ; when it is back from machine B to machine A, it will take zero time. And there is an unlimited buffer on each machine to store jobs that will be processed. The objective is to minimize the completion time on machine B. The complexity of the problem is open in Journal of Scheduling 2001 [5]. We prove the problem is NP-hard in the strong sense by 4-partition problem.

1. Introduction

Flow shop scheduling problem is a classical problem in the OR research field. In this paper, we consider a flow shop scheduling problem with two machines and a transporter. In this problem, we're given a set of n jobs, two machines, A and B and a transporter. Each job has to be processed on machine A first, then transported by a vehicle to machine B finally processed on machine B The interstage transporter can carry at most two jobs between the machines at a time. When the transport carries items from machine A to machine B, it will take time τ; when it is back from machine B to machine A, it will take time σ. And there is an unlimited buffer on each machine to store jobs that will be processed. The objective is to minimize the completion time on machine B. The shortest time is noted by $C_{\max}(S)$, in which S denotes a feasible schedule.

Transporting time is not considered in the classics flow shop scheduling problem. It is assumed that the jobs can be moved between the machines instantaneously. According to the notation that is mentioned in the article written by Lawler et al. in 1993, the classical two-machine flow shop problem is denoted by $F2\|C_{\max}$, where "F" denotes a flow shop problem, and "2" denotes that there are two machines, and "$C_{\max}$" denotes the objective [4]. $F2\|C_{\max}$ problem can be solved within the time of $O(n \log n)$ [2]. Our problem is denoted by $TF2|v=1, c=2|C_{\max}$, where v denotes the number of transporters and c denotes the number of jobs that a transporter can transport in a shipment.

Previous research: Maggu and Das [6] first studied the flow shop with transportation time and proved the problem is solvable if there are a sufficient numbers of transporters. Kise [3] proved that $TF2|v=1, c=1|C_{\max}$ is NP-hard in an ordinary sense by Partition problem and gave other solvable cases. Hurink et al. proved the problem $TF2|v=1, c=1|C_{\max}$ is a NP-hard problem in the strong sense when $\sigma=0$ in 2001 [1]. Lee and Chen further proved that the problem $TF2|v=1, c \geq 3|C_{\max}$ also is a NP-hard problem in the strong sense when $\tau=\sigma$ in 2001. Hall [11] first proposed a polynomial time approx-

* Corresponding author.
E-mail addresses: lanyan@neusoft.edu.cn (Y. Lan), hanxin@dlut.edu.cn (X. Han), wuzongtao813@163.com (Z. Wu), guohe@dlut.edu.cn (H. Guo), chenx_dlut@163.com (X. Chen).

1 Partially supported by NSFC (11571060), RGC (HKU716412E) and "the Fundamental Research Funds for the Central Universities" (DUT15LK10).

http://dx.doi.org/10.1016/j.ipl.2015.09.003
0020-0190/

imation scheme (PTAS) for flow shop problem $F_m\|C_{max}$, i.e., $(1+\epsilon)$-approximation algorithm where F_m means that there are m flow shop machines and m is a constant. The technique in [11] can be directly applied to problem $TF2|v=1, c=1|C_{\max}$ with $\sigma=0$, i.e, there is a polynomial time approximation scheme (PTAS) for $TF2|v=1, c=1|C_{\max}$ with $\sigma=0$. For the model $TF2|v=1, c\geq n|C_{\max}$, Lee and Strusevich [15] obtained a 1.5-approximation algorithm with at most two batches Soper and Strusevich [16] obtained a 4/3-approximation algorithm with at most three batches.

There are several related models: one is the open shop problem with transportation times, where "open shop" means that the order of processing operations of any job is decided by the scheduler. Lushchakova [17] and others obtained a 1.4-approximation algorithm for the problem $TO2|v=1, c\geq n|C_{\max}$.

The other model is called the open shop problem with time lags, where "time lags" means that there is a delay τ_j between the completion time of one operation of job j and the start time of the other operation of job j. This problem is related to problem in [6]. Strusevich [9] gave a 1.5-approximation algorithm for an open shop problem with two machines, where time lag τ_j is job-dependent. If the time lag is job independent, Rayward-Smith and Rebaine proved the problem is a weakly NP-hard problem [13]. Rebaine and Strusevich obtained 1.5-approximation algorithm [14]. For more other models, refer to the survey [10,12].

Our contribution: We prove that the problem $TF2|v=1, c=2|C_{\max}$ with $\sigma=0$ is a NP-hard problem in the strong sense by 4-partition problem.

2. Preliminary

There are machines A and B and a transporter V. There are n jobs, each job has to be processed on machine A first, then transported by V to machine B finally processed on machine B. The transporter is located on machine A in the beginning. Each time, transporter V can carry at most two jobs between the machines. When the transport carries items from machine A to machine B, it will take time τ; when it is back from machine B to machine A, it will take zero time. There is an unlimited buffer on each machine to store jobs that will be processed. The objective is to minimize the completion time on machine B. The shortest time is denoted by $C_{\max}(S)$, in which S denotes a feasible schedule.

Notations: Let S_i^V be the time that job i starts to be transported by vehicle V, let C_i^V be the time when job i is transported to machine B, we have $C_i^V - S_i^V = \tau$. Let S_i^A be the time that job i starts to be processed on machine A, let C_i^A be the time that job i finishes to be processed on machine A, we have $C_i^A - S_i^A = A(i)$, where $A(i)$ is the time that job i is processed on machine A. Let S_i^B be the time that job i starts to be processed on machine B, let C_i^B be the time that job i finishes to be processed on machine B, we have $C_i^B - S_i^B = B(i)$, where $B(i)$ is the time that job i is processed on machine B.

Permutation: A special type of flow shop scheduling is the permutation flow shop scheduling in which the processing order of the jobs on the machines or vehicle is the same for each subsequent step of processing.

Since our problem can be viewed as a special case of classical Flow Shop problem with three machines, the following lemma also holds for our problem.

Lemma 1. *(See [8].) For the classic flow shop problem with three machines, there exists an optimal scheduling which is permutation.*

4-partition: Given $4m$ positive integers $a_1, \ldots, a_{4m}$ with $\sum_{i=1}^{4m} a_i = mb$ and $b/5 < a_i < b/3$, where $i = 1, \ldots, 4m$, does there exist a partition $I_1, \ldots, I_m$ of the index set $\{1, \ldots, 4m\}$ such that $|I_j| = 4$ and $\sum_{i\in I_j} a_i = b$ for $j = 1, \ldots, m$?

The above 4-partition problem is a NP-hard problem in the strong sense [7].

3. NP-hardness of $TF2|v=1, c=2|C_{\max}$

We prove that our problem $TF2|v=1, c=2|C_{\max}$ is a NP-hard problem in strong sense by 4-partition problem, i.e., given an instance of 4-partition, we construct an instance of our problem and prove that if and only if the instance of 4-partition has a solution then there exists a schedule which can finish all the jobs by time y, where y is a given parameter.

Constructing an instance: Given a 4-partition instance $a_1, a_2, \ldots, a_{4m}$, we construct an instance of our problem $TF2|v=1, c=2|C_{\max}$ with $n = 6m+2$ jobs: there are $m+1$ *big* jobs U_j, $4m$ *small* jobs V_j, and $m+1$ *empty* jobs E_j. The construction is as below:

$$
\begin{array}{ll}
\text{big job } U_0: & A(U_0)=1, \quad B(U_0)=2b, \\
\text{big jobs } U_j: & A(U_j)=2b, \\
 & B(U_j)=2b, \ \text{for } j=1,\ldots,m-1, \\
\text{big job } U_m: & A(U_m)=2b, \quad B(U_m)=1, \\
\text{small jobs } V_j: & A(V_j)=a_j, \\
 & B(V_j)=a_j, \ \text{for } j=1,\ldots,4m, \\
\text{empty jobs } E_j: & A(E_j)=0, \\
 & B(E_j)=0, \ \text{for } j=0,\ldots,m.
\end{array}
$$

Let $\tau = b$ and $y = 2 + (3m+1)b$. We are asked whether there is a schedule S such that $C_{\max}(S) \leq y$.

1. *4partition* $\rightarrow$ *TF*2: Assume that $I_1, \ldots, I_m$ is a solution of 4-partition. Then we define a permutation schedule S as shown in Fig. 1. In Machines A, B and the transporter we obey the same order to process all the jobs. The order is that: big job U_0, empty job E_0, four small jobs corresponding to partition I_1, namely $I_1^{(1)}$, $I_1^{(2)}$, $I_1^{(3)}$ and $I_1^{(4)}$. Then big job U_1, empty job E_1, four small jobs corresponding to partition I_2, namely $I_2^{(1)}$, $I_2^{(2)}$, $I_2^{(3)}$ and $I_2^{(4)}$, and so on. When job E_0 finishes at machine A, transporter V carries big job U_0 and empty job E_0 together, arrives at machine B at time $1+b$, then machine B begins to process the two jobs. At time $1+b$, machine A just finishes the four jobs $I_1^{(1)}$, $I_1^{(2)}$, $I_1^{(3)}$ and $I_1^{(4)}$ in I_1, and transporter V is back to machine A and can carry the four jobs in I_1. During time $[1+b, 1+3b]$, transporter V can finish to carry jobs in I_1 by two shipments to machine B. And machine B just finishes the two jobs of big job U_0 and empty job E_0 at time