



# Two-Machine Flow Shop Scheduling Problem Under Linear Constraints

Kameng Nip<sup>1(✉)</sup> and Zhenbo Wang<sup>2</sup>

<sup>1</sup> School of Mathematics (Zhuhai), Sun Yat-sen University, Zhuhai, China  
niejm3@mail.sysu.edu.cn

<sup>2</sup> Department of Mathematical Sciences, Tsinghua University, Beijing, China

**Abstract.** We introduce a two-machine flow shop scheduling problem under linear constraints (2-FLC problem in short), in which the processing times of two stages of jobs are also decision variables and satisfy a system of linear constraints. The goal is to determine the processing times of each job, and to schedule the jobs to the two-machine flow shop such that the makespan, i.e., the completion time of all the jobs is minimized. This problem can find applications in various areas, such as industrial production and advertising planning. We study the computational complexity and algorithms for the 2-FLC problem. Particularly, we show that although the two-machine flow shop scheduling problem can be solved in polynomial time, the 2-FLC problem is generally NP-hard in the strong sense. Then we consider the design and analysis of algorithms on various settings of the 2-FLC problem. In particular, we propose a polynomial time algorithm for the 2-FLC problem when there is a fixed number of constraints. For the general case, we first propose a simple 2-approximation algorithm, and then design a polynomial time approximation scheme (PTAS).

**Keywords:** Flow shop scheduling · Computational complexity · Linear programming · Approximation algorithm

## 1 Introduction

This paper studies a generalization of the two-machine flow shop scheduling problem, in which the processing times of jobs are not given in advance, but can be determined by a system of linear constraints. Given a set of jobs and two machines, each job must be executed on the first machine, and then executed on the second machine, without interruption. The classical problem is to find the schedule which has minimum makespan, i.e., the completion time of the last job. It is usually denoted by  $F2||C_{\max}$  [1]. In our problem, the processing times of jobs are not fixed and exogenously given, but are decision variables that satisfy a set of linear constraints. We call the problem a two-machine flow shop scheduling problem under linear constraints, and 2-FLC for short. The goal of this problem

---

This work has been supported by NSFC No. 11801589, No.11771245 and No. 11371216.

© Springer Nature Switzerland AG 2019

Y. Li et al. (Eds.): COCOA 2019, LNCS 11949, pp. 400–411, 2019.

[https://doi.org/10.1007/978-3-030-36412-0\\_32](https://doi.org/10.1007/978-3-030-36412-0_32)

is to determine the processing times of the jobs, as well as finding the schedule that has minimum makespan among all the feasible choices.

The current work can be viewed as an extension of the existing works on the scheduling problem under linear constraints [2, 3]. In these works, several single-stage scheduling problems under linear constraints, including parallel machine scheduling [2], related machine scheduling [3] have been introduced and studied. Under this framework, some parameters of the scheduling problem, such as the processing times or machine speeds must satisfy a set of linear constraints. The scheduling problem under linear constraints can be simpler or harder than the original scheduling problem. These problems may have different computational complexities, and require different techniques in designing and analyzing algorithms. For instance, the parallel machine scheduling where the processing times satisfying linear constraints [2] can be solved in polynomial time if both the number of constraints and machines are fixed constants, whereas the parallel machine scheduling problem itself is NP-hard even when there are only two machines. These results suggest us to explore various scheduling models, and to study their complexities and design of algorithms under linear constraints. Apart from scheduling problems, there are also some other researches on various combinatorial optimization problems under linear constraints in the literature, such as bin packing problem [4] and knapsack problem [5].

Next we describe some application scenarios that motivate the study of our 2-FLC problem. We remark that the scenarios are actually the extensions of those introduced in [2, 3]. In fact, we can naturally replace the machine requirement from parallel machines to flow-shop machines in their examples.

1. *Industrial Production Problem.* Consider a problem arises in the steel industry, in which the decision maker requires certain amounts of different raw metals (say, iron, copper, aluminum, and etc.). The raw metals are obtained by extracting from several different types of steel. The extraction of each type of steel can be seen as a job. This is a typical application of flow shop scheduling problem on the industrial production (see, e.g., [6]). For example, each job should undergo wire-drawing first, and then annealing, which can be regarded as the two stages of flow-shop machines. The goal is to find a schedule of the jobs. e.g., to finish the entire production as early as possible. In practice, the processing time of each job depends on the processing quantities of steel that needs to be processed. Moreover, those quantities are usually determined by the demands of the raw metals and can be formulated as a blending problem [7, 8]. Table 1 is a concrete example.

Let  $x_i$  be the processing time of steel  $i$  to be processed in the first machine (e.g., wire-drawing), and  $y_i$  be the processing time of steel  $i$  to be processed in the second machine (e.g., annealing). In the example shown in Table 1, the demand of iron is 56. If we assign  $x_1 + y_1$  units of processing time on steel 1 in total, then we can produce  $24(x_1 + y_1)$  unit of iron. Similarly, if we assign  $x_2 + y_2$  units of processing time on steel 2 in total, then we can produce  $8(x_2 + y_2)$  unit of iron. Then the requirement on the demand of iron can be represented as a linear inequality

$24(x_1 + y_1) + 8(x_2 + y_2) + 3(x_3 + y_4) + \dots + 2(x_n + y_n) \geq 56$ . Moreover, the processing time of a job in the second stage always depend on the processing time and (amounts of steel) first stage. For example, if the processing time (amounts) of steel 1 the first stage is  $x_1$  units, then we require exactly  $x_1/2$  units of processing time on the second machine, which can be represented as a linear equality  $x_1 = 2y_1$ . Due to the environment of the machines, there are usually some limits on the processing times of the steels on both machines. For example, the maximum processing time of steel 1 on the first machine is 10, and the processing time of steel 1 on the second machine should be within  $[2, 7]$ . Therefore, we have constraints  $x_1 \leq 10$  and  $5 \leq y_1 \leq 7$ . Similarly, we can write linear constraints for the demand of other required metals and other constraints. In this problem, the decision maker needs to determine the non-negative job processing times  $x_1, \dots, x_n, y_1, \dots, y_n$  satisfying the above linear constraints, and then assign these jobs to the two-machine flow shop machines such that the last completion time is minimized. This problem can be viewed as a minimum makespan two-machine flow shop scheduling problem, where the processing times of jobs satisfy several linear constraints.

2. *Advertising Media Production and Planning Problem.* The flow shop scheduling problem can find application in the area of media and Internet (see, e.g., [9, 10]). For example, deciding the sequences of the media production and broadcast can be regarded as two stages of the flow shop machine scheduling problem. Consider a scenario that a media production company has several potential advertisements that need to be produced and then be broadcast. Let  $x_i$  be the entire production time (say, e.g., casting, filming and post production) and  $y_i$  be the duration of the advertisement  $i$ . Each advertisement  $i$  must be broadcast after its production is finished, and the goal is to finish the entire broadcast of all the advertisements as early as possible. The whole process can be viewed as a two-machine flow shop scheduling problem. Furthermore, the decision maker is required to decide the production time and duration for the advertisements, that is, the processing times of the jobs. It can always be represented as several linear constraints. An example of such a problem is given in Table 2.

In Table 2, there is a budget of the production of all advertisements. Each unit production time of advertisement 1 costs 100, each unit production time of advertisement 2 costs 200, and so forth. The total budget constraint is no more than 10000, and hence can be represented as  $100x_1 + 200x_2 + \dots + 150x_n \leq 10000$ . Moreover, the company will gain revenue  $2000y_1$  for  $y_1$  units of broadcasting time of advertisement 1. The company requires a total revenue at least 15000, which can be represented as  $1000x_1 + 2000x_2 + \dots + 1500x_n \geq 15000$ . The production time always depending on the duration (broadcast time) of the advertisement. For example, each unit of broadcast time of advertisement 1 requires at least 10 units of production time, which can be represented as  $x_1 \geq 10y_1$ . Similar to the first example, we can formulate the other constraints as several linear constraints. The above-described problem can be naturally

**Table 1.** Example for the industrial production problem

Composition	Steel				Demand
	1	2	3	... n	
Iron	$24(x_1 + y_1)$	$8(x_2 + y_2)$	$3(x_3 + y_3)$	$\cdots 2(x_n + y_n)$	$\geq 56$
Copper	$3(x_1 + y_1)$	$3(x_2 + y_2)$	$3(x_3 + y_3)$	$\cdots 1(x_n + y_n)$	$\geq 30$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Aluminum	$4(x_1 + y_1)$	$33(x_2 + y_2)$	$13(x_3 + y_3)$	$\cdots 100(x_n + y_n)$	$\geq 100$
Relations					
Steel 1	$x_1 - 2y_1$	0	0	$\cdots 0$	$= 0$
Steel 2	0	$2x_2 - y_2$	0	$\cdots 0$	$= 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Steel n	0	0	0	$\cdots x_n - 10y_n$	$= 0$
Limits					
Max. of 1(Stage 1)	$x_1$	0	0	$\cdots 0$	$\leq 10$
Min. of 1(Stage 2)	$y_1$	0	0	$\cdots 0$	$\geq 2$
Max. of 1(Stage 2)	$y_1$	0	0	$\cdots 0$	$\leq 7$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Max. of n (Stage 2)	0	0	0	$\cdots y_n$	$\leq 15$

formulated as a two-machine flow shop scheduling problem in which the processing times (running times of the advertisements) are determined by a system of linear constraints.

In this paper, we study the computational complexity and algorithms for the 2-FLC problem. First, we show that the problem is generally NP-hard in the strong sense. It is well-known that the original two-machine flow shop scheduling problem can be solved in  $O(n \log n)$  time by Johnson’s rule [11], where  $n$  is the number of jobs. This result suggests that the 2-FLC problem with arbitrary number of constraints has a very different complexity from the original two-machine flow shop scheduling problem. Then we consider the design of algorithms on various settings of the 2-FLC problem. In particular, we design a polynomial time algorithm for the 2-FLC problem when there is a fixed number of constraints. For the general case, we first propose a simple 2-approximation algorithm, and then design a polynomial time approximation scheme (PTAS).

The remainder of this paper is organized as follows: In Sect. 2, we give a formal definition of the 2-FLC problem studied in this paper, and briefly review some related literature. In Sect. 3, we study the computational complexity of the 2-FLC problem. In Sect. 4, we discuss the case with a fixed number of constraints. In Sect. 5, we study the approximation algorithm for the general case. We provide some concluding remarks in Sect. 6.

**Table 2.** Example for the advertising media production and planning problem

	Each unit time broadcast provides					
	ad 1	ad 2	ad 3	...	ad $n$	
Production budget	$100x_1$	$200x_2$	$100x_3$	...	$150x_n$	$\leq 10000$
Broadcast revenue	$2000y_1$	$3500x_2$	$2000y_3$	...	$1500y_n$	$\geq 150000$
Attractions to women	$20y_1$	$100y_2$	$100y_3$	...	$10y_n$	$\geq 500$
Attractions to men	$15y_1$	$10y_2$	$0y_3$	...	$80y_n$	$\geq 500$
Attractions to teens	$30y_1$	$0y_2$	$30y_3$	...	$100y_n$	$\geq 200$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Relation of ad 1	$x_1 - 10y_1$	0	0	...	0	$\geq 0$
Relation of ad 2	0	$x_2 - 15y_2$	0	...	0	$\geq 0$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Max time for ad 1	$y_1$	0	0	...	0	$\leq 20$
Min time for ad 1	$y_1$	0	0	...	0	$\geq 10$
Max time for ad 2	0	$y_2$	0	...	0	$\leq 35$
⋮	⋮	⋮	⋮	⋮	⋮	⋮

## 2 Problem Description

In this paper, we consider the following the two-machine flow shop scheduling problem under linear constraints (2-FLC) problem stated below:

**Definition 1.** *There are  $n$  jobs and two flow-shop machines. Each job has to be processed on the first machine and then on the second machine without interruption. The processing times of job  $i$  on the first machine and the second machine are  $x_i$  and  $y_i$  respectively, which are determined by  $k$  linear constraints. The goal of the two-machine flow shop scheduling under linear constraints (2-FLC) problem is to determine the processing times of the jobs such that they satisfy the linear constraints and to schedule the jobs to the two flow-shop machines such that the makespan is minimized.*

Let  $A, C \in \mathbb{R}^{k \times n}$ ,  $\mathbf{b} \in \mathbb{R}^{k \times 1}$ , the processing times  $\mathbf{x} = \{x_1, \dots, x_n\}$ ,  $\mathbf{y} = \{y_1, \dots, y_n\}$  should be feasible solutions to  $A\mathbf{x} + C\mathbf{y} \geq \mathbf{b}$ ,  $\mathbf{x}, \mathbf{y} \geq 0$ . It is well-known that the two-machine flow shop scheduling problem  $F2||C_{\max}$  has an optimal schedule which is a permutation schedule (see, e.g., [9]), i.e., the orders of jobs in both machines are identical. Therefore, we can denote a schedule of our problem as  $\sigma = \{\sigma(1), \dots, \sigma(n)\}$ , where  $\sigma(i)$  indicates the job that is assigned on position  $i$  in the schedule. The makespan  $C_{\max}$  of a schedule is thus the completion time of job  $\sigma(n)$  on the second machine.

Here we give a brief literature review on the flow shop scheduling and related problems. Flow shop scheduling is one of the three basic models (open shop, flow

shop, job shop) of multi-stage scheduling problems. The flow shop scheduling which minimizes the makespan is usually denoted by  $Fm||C_{\max}$ , where  $m$  is the number of machines. Garey et al. proved that  $Fm||C_{\max}$  is strongly NP-hard for  $m \geq 3$  [12], and Hall proposed a PTAS algorithm for  $Fm||C_{\max}$  [13]. Particularly, the two-machine flow shop scheduling problem  $F2||C_{\max}$  can be solved by Johnson’s algorithm in  $O(n \log n)$  time [11]. If all the jobs are processed in the same order, then we call this schedule a permutation schedule. It has been shown in [14] that  $F2||C_{\max}$  or  $F3||C_{\max}$  has an optimal permutation schedule.

### 3 Computational Complexity of 2-FLC Problem

We first study the computation complexity of 2-FLC problem. We show that this problem is NP-hard in the strong sense, even though the original  $F2||C_{\max}$  problem can be solved in polynomial time by Johnson’s algorithm.

**Theorem 1.** *The 2-FLC problem with two machines is NP-hard in the strong sense.*

*Proof.* We reduce the Exact Cover by 3-Sets problem (X3C) to the 2-FLC problem. Given a ground set  $S$  with  $|S| = 3n$  ( $n \geq 1$ ) and a collection  $\mathcal{C}$  of 3-element subsets of  $S$ . The X3C problem is to decide if there exists a subset  $\mathcal{C}'$  of  $\mathcal{C}$  with size  $n$ , in which every element of  $S$  occurs in  $\mathcal{C}'$  exactly once. Let  $m = |\mathcal{C}|$  be the total number of subsets and assume that  $m > n$  without loss of any generality. We construct an instance of 2-FLC with  $m + 1$  jobs. Each subset  $S_i \in \mathcal{C}$  is associated with a job  $i$  that has processing times  $x_i$  and  $y_i$  in the two stages, respectively. The remaining job  $m + 1$  has processing times  $x_{m+1} = n$  in the first stage and  $y_{m+1} = m - n$  in the second stage. The processing times are determined by the following linear constraints:

$$\sum_{i:j \in S_i} y_i = 1 \qquad \forall j \in S \qquad (1a)$$

$$x_j = 1 - y_j \qquad \forall i = 1, \dots, m \qquad (1b)$$

$$x_{m+1} = n, \quad y_{m+1} = m - n \qquad (1c)$$

$$\mathbf{x}, \mathbf{y} \geq 0.$$

We show that the instance for X3C is YES if and only if there exists a schedule of 2-FLC in which the processing times are feasible to (1) and has makespan at most  $m$ .

First, suppose that there exist an exact cover  $\mathcal{C}'$  of  $S$ . Let  $x_i = 0, y_i = 1$  if  $S_i \in \mathcal{C}'$  and  $x_i = 1, y_i = 0$  otherwise, and  $x_{m+1} = n, y_{m+1} = m - n$ . Since  $\mathcal{C}'$  is an exact cover, it can be seen that the linear constraints (1) are satisfied. By Johnson’s rule, the jobs correspond to  $S_i \in \mathcal{C}'$  are scheduled first, then the job  $m + 1$ , and last the jobs correspond to  $S_i \notin \mathcal{C}'$ . Note that the number of subsets in every exact cover is  $n$ , therefore the makespan of the schedule is exactly  $n + (m - n) = m$ .

Now we prove the opposite direction. Note that the job  $m + 1$  has total processing time  $n + (m - n) = m$ , hence the makespan of this schedule is exactly  $m$ . The constraints in (1b) guarantees that all the  $x_i$ s and  $y_i$ s are binary, as otherwise the makespan of the schedule must be more than  $m$ . To see this, if there are some job  $k$  which has processing time  $x_k \geq y_k > 0$  (the case with  $0 < x_k < y_k$  is analogous), then job  $k$  must be assigned after job  $m + 1$  by Johnson's rule, which leads to a schedule with makespan at least  $n + (m - n) + y_k > m$ . This contradicts to the condition that the schedule has makespan at most  $m$ . Denote  $J'$  as the jobs having processing time  $x_i = 0, y_i = 1$  and  $\bar{J}$  as the remaining jobs, that is, the jobs have processing time  $x_i = 1, y_i = 0$ . Again, by Johnson's rule and the makespan is  $m$ , it can be seen that there is exactly  $n$  jobs in  $J'$ . By (1a), the subsets in  $\mathcal{C}$  corresponds to  $J'$  constitute an exact cover of  $S$ .  $\square$

### 4 Fixed Number of Constraints

In this section, we study the case when the number of constraints is fixed.

**Lemma 1.** *The 2-FLC problem has an optimal solution, in which each machine has at most  $k$  jobs with nonzero processing time.*

*Proof.* We first show that given any optimal solution to the 2-FLC problem, we can construct an optimal solution that at most  $k$  jobs have nonzero processing time in the second machine. Now we fix the permutation of jobs in an optimal solution. Note that the jobs have same orders in both machines. To simplicity of notation, we denote the order of the jobs as  $1, \dots, n$ . We can construct the following linear program:

$$\min \quad t \tag{2a}$$

$$\text{s.t.} \quad \sum_{i=1}^l x_i + \sum_{i=l}^n y_i \leq t \quad \forall l = 1, \dots, n \tag{2b}$$

$$\sum_{i=1}^n a_{ji}x_i + c_{ji}y_i \geq b_j \quad \forall j = 1, \dots, k \tag{2c}$$

$$t, x_i, y_i \geq 0 \quad \forall i = 1, \dots, n.$$

It can be seen that any optimal solution to (2) is also optimal to the 2-FLC problem. To see this, let  $(\mathbf{x}^*, \mathbf{y}^*)$  be the processing times of the optimal solution to the 2-FLC problem. Then by the  $(\mathbf{x}, \mathbf{y})$  is the optimal solution to (2), we have  $t = \max_{l=1, \dots, n} \left\{ \sum_{i=1}^l x_i + \sum_{i=l}^n y_i \right\} \geq \left\{ \sum_{i=1}^l x_i^* + \sum_{i=l}^n y_i^* \right\} = C_{\max}^*$ , where  $C_{\max}^*$  is the optimal makespan to the 2-FLC problem.

Introducing slack variables to (2b), we obtain a system of  $n$  constraints:

$$\sum_{i=1}^l x_i + \sum_{i=l}^n y_i + z_i = t \quad \forall l = 1, \dots, n. \tag{3}$$

Applying Gaussian elimination to (3), we obtain

$$\begin{aligned}
 x_1 &= t - \sum_{i=1}^n y_i - z_1 & (4) \\
 x_i &= y_{i-1} + z_{i-1} - z_i \quad \forall i = 2, \dots, n.
 \end{aligned}$$

Therefore, (2a) is equivalent to the following linear program:

$$\min \quad t \tag{5a}$$

$$\text{s.t. } x_1 + z_1 = t - \sum_{i=1}^n y_i \tag{5b}$$

$$x_i + z_i = y_{i-1} + z_{i-1} \quad \forall i = 2, \dots, n \tag{5c}$$

$$\sum_{i=1}^n a_{ji}x_i + c_{ji}y_i \geq b_j \quad \forall j = 1, \dots, k \tag{5d}$$

$$t, x_i, y_i, z_i \geq 0 \quad \forall i = 1, \dots, n,$$

Now we find the optimal basic feasible solution to (5), which is also an optimal solution to the 2-FLC problem. Note that there are at most  $n + k$  variables among  $x_i, y_i, z_i$ , and  $t$  can have nonzero processing time. If  $t = 0$ , then all  $x_i$ s and  $y_i$ s are zero and the lemma is proved. Therefore, we assume that  $t > 0$  and thus at most  $n + k - 1$  of  $x_i, y_i$ , and  $z_i$  have nonzero processing time. Let  $S = \{i \in \{2, \dots, n\} \mid x_i + z_i > 0\}$  be the subset of indices that with  $x_i + z_i$  is positive. If  $|S| = n - 1$ , then at most  $k$  of the  $y_i$ s have nonzero processing times, since the total number of remaining variables which can have nonzero processing times it at most  $n + k - 1 - (n - 1) = k$ . Otherwise,  $|S| < n - 1$  and there exists some  $i$  such that  $x_i + z_i = 0$ . Note that by constraint (5c), the corresponding  $y_{i-1} = z_{i-1} = 0$ . Now we fix the processing times of all those  $x_i, z_i$  and  $y_{i-1}$  to be zero, and consider the linear program without these variables:

$$\min \quad t \tag{6a}$$

$$\text{s.t. } x_1 + z_1 = t - \sum_{i \in S} y_{i-1} \tag{6b}$$

$$x_i + z_i = y_{i-1} + z_{i-1} \quad \forall i \in S \tag{6c}$$

$$\sum_{i \in \{1\} \cup S} a_{ji}x_i + \sum_{i \in S} c_{j,i-1}y_{i-1} \geq b_j \quad \forall j = 1, \dots, k \tag{6d}$$

$$t, x_1, z_1, x_i, y_{i-1}, z_i \geq 0 \quad \forall i \in S.$$

Note that the optimal solution to (5) is feasible to (6), hence the optimal solution to (6) is still optimal to 2-FLC. We repeat the above procedure, that is, find an optimal basic feasible solution to (6), which has at most  $|S| + k$  variables with nonzero processing time. If at some point  $x_i + z_i$  is strictly greater than zero for all  $i \in S$ , then we obtain an optimal basic feasible solution in which at most  $|S| + k - |S| = k$  of the  $y_i$ s have nonzero processing times; otherwise we can reduce



the size of  $|S|$  and find the optimal basic feasible solution to the reduced linear program similar to (6), and repeat the above procedure. It can be observed that the size of  $S$  is strictly decreasing. Therefore if the procedure continues, we must have  $|S| \leq k$  at some point. In that case, we obtain an optimal basic feasible solution which has at most  $k$  of the  $x_i + z_i$ s have nonzero processing times (note that the  $x_i + z_i$  for all  $i \notin S$  have been fixed to be zero before), and so for the  $x_i$ s and  $y_i$ s. Therefore, there exists an optimal solution that the second machine must have at most  $k$  nonzero processing time jobs. Fixed the processing times on the second machine and applied the same method on the first machine, we obtain a solution that each machine has at most  $k$  nonzero processing time jobs.  $\square$

Based on Lemma 1, we can propose a polynomial time algorithm for the 2-FLC problem when the number of constraints is a fixed number of constraints. We summarize the algorithm as Algorithm 1 and Theorem 2.

---

**Algorithm 1.** Enumeration algorithm for 2-FLC problem with fixed  $k$

---

**Input:**  $n$  jobs,  $k$  linear constraints  $A\mathbf{x} + C\mathbf{y} \geq \mathbf{b}$

**Output:** the schedule to the two flow-shop machines that has the minimum makespan, and the corresponding processing times of jobs

- 1: **for** each subset  $J'$  of  $J$  with  $k$  jobs **do**
- 2:   **for** each permutation of the jobs in  $J'$  **do**
- 3:     Let  $\sigma(1), \dots, \sigma(k)$  be the permutation, i.e., the schedule of these  $k$  jobs. Solve the following LP while setting  $y_i = 0$  for  $i \notin J'$ .

$$\min \quad t \tag{7a}$$

$$\text{s.t.} \quad \sum_{i=1}^l x_{\sigma(i)} + \sum_{i=l}^k y_{\sigma(i)} \leq t \quad \forall l = 1, \dots, k \tag{7b}$$

$$\sum_{i=1}^n x_i \leq t \tag{7c}$$

$$\sum_{i=1}^n a_{ji}x_i + c_{ji}y_i \geq b_j \quad \forall j = 1, \dots, k \tag{7d}$$

$$y_i = 0 \quad \forall i \notin J' \tag{7e}$$

$$t, x_i, y_i \geq 0 \quad \forall i = 1, \dots, n.$$

- 4:   **if** (7) is infeasible **then**
  - 5:     Let the processing times of jobs be the optimal solution to (7), and record the schedule and the makespan.
  - 6: **return** the schedule with the smallest makespan among all these iterations and its corresponding processing times.
- 

**Theorem 2.** *The 2-FLC problem when the number of constraints is fixed has a polynomial time algorithm with complexity  $O(n^{k+3}L)$ .*

The proofs of the subsequent lemmas/theorems will be provided in Appendix.

## 5 General Case: Approximation Algorithms

First we propose a simple 2-approximation algorithm for the problem. We solve the linear program with minimum total progressing time subject to  $Ax = b$  to obtain the processing times  $\mathbf{x}, \mathbf{y}$ . Then we assign the jobs with processing times  $\mathbf{x}, \mathbf{y}$  by Johnson’s rule.

**Theorem 3.** *There is a 2-approximation algorithm for the 2-FLC problem.*

Now we propose a PTAS for the problem. First, we run the 2-approximation algorithm and obtain a value  $C_{\max}$ . By Theorem 1,  $C_{\max}/2$  and  $C_{\max}$  are lower bound and upper bound of the optimal makespan of the 2-FLC problem, respectively. We denote  $LB = C_{\max}/2$  and  $UB = C_{\max}$ .

Given  $\epsilon > 0$ , we denote the job sets (with respect to  $\mathbf{x}$  and  $\mathbf{y}$ ) as follow:

$$\begin{aligned} J_1 &= \{i \in J \mid x_i \leq y_i\}, & L_1 &= \{i \in J \mid x_i \leq y_i, y_i > \epsilon C_{\max}^*(\mathbf{x}, \mathbf{y})\}, \\ J_2 &= \{i \in J \mid x_i \geq y_i\}, & L_2 &= \{i \in J \mid x_i \geq y_i, x_i > \epsilon C_{\max}^*(\mathbf{x}, \mathbf{y})\}, \end{aligned} \tag{8}$$

where  $C_{\max}^*(\mathbf{x}, \mathbf{y})$  is the makespan of optimal schedule of  $F2||C_{\max}$  with processing time  $\mathbf{x}$  and  $\mathbf{y}$ . Note that we have the sizes of  $L_1$  and  $L_2$  are  $|L_1| \leq 1/\epsilon$  and  $|L_2| \leq 1/\epsilon$  as otherwise we have  $\sum_{i=1}^n x_i > C_{\max}^*(\mathbf{x}, \mathbf{y}) \geq C_{\max}^*$  or  $\sum_{i=1}^n y_i > C_{\max}^*(\mathbf{x}, \mathbf{y}) \geq C_{\max}^*$ , which leads to a contradiction. The algorithm first guesses the value of the optimal makespan, then guesses the set  $L_1$  and  $L_2$  in an optimal solution (w.r.t.  $\mathbf{x}^*$  and  $\mathbf{y}^*$ ), as well as enumerating all the possible permutations of these jobs. During each guess, we obtain the values of the processing times of jobs by checking the feasibility of a specific linear program. The jobs are then scheduled by the Johnson’s rule. The algorithm finally returns the best schedule among all these iterations. We summarize the details as Algorithm 2.

To establish the approximation performance result of Algorithm 2, we first state a property of a specified schedule in the classical two-machine flow shop scheduling problem  $F2||C_{\max}$ , which has makespan at most  $1 + \epsilon$  of the optimal solution to  $F2||C_{\max}$  and allows us to design a PTAS for the 2-FLC problem. Fixed any processing times  $\mathbf{x}$  and  $\mathbf{y}$  for the jobs,  $J_1, J_2, L_1, L_2, C_{\max}^*(\mathbf{x}, \mathbf{y})$  are defined as (8),  $S_1 = J_1 \setminus L_1 = \{i \in J \mid x_i \leq y_i, y_i \leq \epsilon C_{\max}^*(\mathbf{x}, \mathbf{y})\}$  and  $S_2 = J_2 \setminus L_2 = \{i \in J \mid x_i \geq y_i, x_i \leq \epsilon C_{\max}^*(\mathbf{x}, \mathbf{y})\}$ . The lemma is described as below.

**Lemma 2.** *Let  $C'_{\max}(\mathbf{x}, \mathbf{y})$  be the makespan of schedule obtained by assigning the jobs in the order  $S_1, L_1, L_2, S_2$  (identical for both machines), where the jobs in  $L_1$  and  $L_2$  are scheduled according to Johnson’s rule,  $S_1$  and  $S_2$  are scheduled arbitrarily. Then,  $C'_{\max}(\mathbf{x}, \mathbf{y}) \leq (1 + \epsilon)C_{\max}^*(\mathbf{x}, \mathbf{y})$ .*

Based on this lemma, we have the following result.

**Theorem 4.** *Algorithm 2 is a PTAS for the 2-FLC problem.*

**Algorithm 2.** PTAS for 2-FLC problem**Input:**  $n$  jobs,  $k$  linear constraints  $A\mathbf{x} + C\mathbf{y} \geq \mathbf{b}$ **Output:** the schedule to the two flow-shop machines that has the minimum makespan, and the corresponding processing times of jobs

- 1: Given  $\epsilon \in (0, 1)$ , let  $h = \lceil 1/\epsilon \rceil$  and  $UB, LB$  defined as before
- 2: **for** each subset  $L_1^h$  and each subset  $L_2^h$  of  $J$ , with  $L_1^h \cup L_2^h = \emptyset$  and  $|L_1^h| = |L_2^h| = h$  **do**
- 3:   **for** each permutation  $\sigma_1 = (\sigma_1(1), \dots, \sigma_1(h))$  and each permutation  $\sigma_2 = (\sigma_2(1), \dots, \sigma_2(h))$  of  $(1, \dots, h)$  **do**
- 4:     Denote  $(\sigma_1(1), \dots, \sigma_1(h))$  and  $(\sigma_2(1), \dots, \sigma_2(h))$  as the jobs and the schedule in  $L_1^h$  and  $L_2^h$ , respectively.
- 5:     Set  $S^h = J \setminus (L_1^h \cup L_2^h)$ ,  $T = LB$ .
- 6:     **while**  $T < (1 + \epsilon)UB$  **do**
- 7:       Check the feasibility of the following linear program (9):

$$A\mathbf{x} + C\mathbf{y} \geq \mathbf{b} \quad (9a)$$

$$\sum_{i=1}^n x_i \leq (1 + \epsilon)T, \quad \sum_{i=1}^n y_i \leq (1 + \epsilon)T \quad (9b)$$

$$\sum_{i \in S^h} t_i + \sum_{i=1}^l x_{\sigma_1(i)} + \sum_{i=1}^h y_{\sigma_1(i)} + \sum_{i=1}^h y_{\sigma_2(i)} \leq (1 + \epsilon)^2 T \quad \forall l \in 1, \dots, h \quad (9c)$$

$$\sum_{i \in S^h} t_i + \sum_{i=1}^h x_{\sigma_1(i)} + \sum_{i=1}^l x_{\sigma_2(i)} + \sum_{i=1}^h y_{\sigma_2(i)} \leq (1 + \epsilon)^2 T \quad \forall l = 1, \dots, h \quad (9d)$$

$$t_i \geq x_i, \quad t_i \geq y_i \quad \forall i = 1, \dots, n \quad (9e)$$

$$y_j \leq y_i, \quad x_i \leq y_i \quad \forall j \in S^h, i \in L_1^h \quad (9f)$$

$$x_j \leq x_i, \quad y_i \leq x_i \quad \forall j \in S^h, i \in L_2^h \quad (9g)$$

$$x_i, y_i, t_i \geq 0, \quad \forall i = 1, \dots, n.$$

- 8:   **if** (9) is infeasible **then**
- 9:     Set  $T = (1 + \epsilon)T$ , continue to Step 6.
- 10:   **else**
- 11:     Record the processing times  $\mathbf{x}$  and  $\mathbf{y}$  and  $T$  of a feasible solution to 9, continue to Step 3.
- 12: **return** the solution with the smallest  $T$  among all these iterations and its corresponding processing times, and schedule the jobs by Johnson's rule.

## 6 Conclusions

We study the two-machine flow shop scheduling problem under linear constraints in this paper. The problem is NP-hard in the strong sense, and we propose several

optimal or approximation algorithms for various settings of it. An immediate research question is to study the FLC problem where the number of machines is more than 2. Furthermore, it is also interesting to consider the shop scheduling problem under linear constraints where the machine environment is open shop or job shop, and the shop scheduling problem with additional constraints, such as the no-wait case.

## References

1. Graham, R., Lawler, E., Lenstra, J., Kan, A.: Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discret. Math.* **5**(1), 287–326 (1979)
2. Nip, K., Wang, Z., Wang, Z.: Scheduling under linear constraints. *Eur. J. Oper. Res.* **253**(2), 290–297 (2016)
3. Zhang, S., Nip, K., Wang, Z.: Related machine scheduling with machine speeds satisfying linear constraints. In: Kim, D., Uma, R.N., Zelikovsky, A. (eds.) *COCOA 2018*. LNCS, vol. 11346, pp. 314–328. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-04651-4\\_21](https://doi.org/10.1007/978-3-030-04651-4_21)
4. Wang, Z., Nip, K.: Bin packing under linear constraints. *J. Comb. Optim.* **34**(5), 1198–1209 (2017)
5. Nip, K., Wang, Z., Wang, Z.: Knapsack with variable weights satisfying linear constraints. *J. Global Optim.* **69**(3), 713–725 (2017)
6. Nagar, A., Heragu, S.S., Haddock, J.: A branch-and-bound approach for a two-machine flowshop scheduling problem. *J. Oper. Res. Soc.* **46**(6), 721–734 (1995)
7. Danø, S.: *Linear Programming in Industry, Theory and Applications: An Introduction*. Springer, Vienna (1960). <https://doi.org/10.1007/978-3-7091-3644-7>
8. Eiselt, H.A., Sandblom, C.L.: *Linear Programming and its Applications*. Springer, Heidelberg (2007). <https://doi.org/10.1007/978-3-540-73671-4>
9. Emmons, H., Vairaktarakis, G.: *Flow Shop Scheduling: Theoretical Results, Algorithms, and Applications*. Springer, New York (2013). <https://doi.org/10.1007/978-1-4614-5152-5>
10. Allahverdi, A., Al-Anzi, F.S.: Using two-machine flowshop with maximum lateness objective to model multimedia data objects scheduling problem for WWW applications. *Comput. Oper. Res.* **29**(8), 971–994 (2002)
11. Johnson, S.M.: Optimal two- and three-stage production schedules with setup times included. *Naval Res. Logistics Q.* **1**, 61–68 (1954)
12. Garey, M.R., Johnson, D.S., Sethi, R.: The complexity of flowshop and jobshop scheduling. *Math. Oper. Res.* **1**, 117–129 (1976)
13. Hall, L.A.: Approximability of flow shop scheduling. *Math. Program.* **82**, 175–190 (1998)
14. Conway, R.W., Maxwell, W., Miller, L.: *Theory of Scheduling*. Addison-Wesley, Reading (1967)