

This article was downloaded by: [National Chiao Tung University 國立交通大學]

On: 24 April 2014, At: 07:31

Publisher: Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



## International Journal of Production Research

Publication details, including instructions for authors and subscription information:  
<http://www.tandfonline.com/loi/tprs20>

### Batch scheduling in differentiation flow shops for makespan minimisation

Ting-Chih Huang<sup>a</sup> & Bertrand M.T. Lin<sup>a</sup>

<sup>a</sup> Department of Information and Finance Management, Institute of Information Management, National Chiao Tung University, Taiwan

Published online: 09 Jun 2013.

To cite this article: Ting-Chih Huang & Bertrand M.T. Lin (2013) Batch scheduling in differentiation flow shops for makespan minimisation, International Journal of Production Research, 51:17, 5073-5082, DOI: [10.1080/00207543.2013.784418](https://doi.org/10.1080/00207543.2013.784418)

To link to this article: <http://dx.doi.org/10.1080/00207543.2013.784418>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

## Batch scheduling in differentiation flow shops for makespan minimisation

Ting-Chih Huang and Bertrand M.T. Lin\*

*Department of Information and Finance Management, Institute of Information Management, National Chiao Tung University, Taiwan*

*(Received 11 November 2012; final version received 4 March 2013)*

This paper considers a two-stage three-machine differentiation flow shop that comprises a common machine at stage 1 and two independent dedicated machines at stage 2. Two types of jobs are to be processed. All jobs must visit the stage-1 machine, and then the jobs of each type proceed to their dedicated machine for stage-2 processing. The stage-1 machine processes the jobs in batches, each of which, whenever formed, requires a constant setup time. The objective is to find a schedule that attains the minimum makespan. While the problem is strongly NP-hard, we investigate the case where the processing sequences of the two types of jobs are given and fixed. A polynomial-time dynamic programming algorithm is designed to solve this problem. We then deploy this algorithm to compute the lower bounds of the general problem.

**Keywords:** flow shop scheduling; mass customization; scheduling; makespan

### 1. Introduction

Delayed differentiation or postponement is a concept in supply chain management where the manufacturing process starts by making a generic or family product that is later differentiated into a specific end-product. This is a widely used method, especially in industries with high demand uncertainty or the need of mass customisation. In particular, the model of delayed differentiation is one of the major approaches to achieving mass customisation (Da Silveira, Borenstein, and Fogliatto 2001; Simchi-Levi, Kaminsky, and Simchi-Levi 2007). Gupta and Benjaafar (2004) also mention that delayed differentiation carries several benefits. Maintaining stocks of semi-finished goods reduces the order-fulfillment delay related to the pure MTO (Make-To-Order) system. Since many different end-products have common parts, holding semi-finished goods inventory benefits from demand pooling, which is known to lower the amount of inventory needed to achieve a service-level performance comparable to that of a system with no pooling (Eppen 1979). Furthermore, investment in semi-finished inventories is smaller when compared with the option of maintaining a similar amount of finished-goods inventory. Additional benefits from delaying differentiation include significantly streamlined MTS (Make-To-Stock) segments of the manufacturing process and simplification of production scheduling, sequencing and raw material purchasing. Examples of the application of delayed differentiation include Benetton's knitwear (Bruce 1987), Dell's computers (Magretta 1998) and HP's printers (Feitzinger and Lee 1997).

In this paper, we link a two-stage supply chain with a two-stage flow shop. When we consider a supply chain in an abstract way, it can be viewed as a production line within an organisation. A flow-shop-type production consists of machines arranged in series such that all products need to be processed in the order that the machines are arranged in. Kyparisis and Koulamas (2000), who study this two-stage flow shop, report that "applications of the proposed flow shop model are encountered in manufacturing settings, where all jobs must first go through the same main process, and then they require a finishing operation special to the job".

Scheduling refers to managerial decision making that allocates limited resources to activities so as to optimise, subject to functional constraints or assumptions, a certain set of performance measures. Scheduling is crucial to operations management of applications in manufacturing and service industries (Pinedo and Chao 1999; Pinedo 2012). Since the seminal work of Johnson (1954), flow shop scheduling has received considerable research attention (Dudek, Panwalkar, and Smith 1992; Reisman, Kumar, and Motwani 1997). This broad topic contains many different settings and special cases, reflecting a wide range of applications. In this paper, we consider a special three-machine two-stage flow shop called a differentiation flow shop, where all products (jobs) share a common critical machine in the primary (first) stage, and then each individual product proceeds to a dedicated machine in the successive (second) stage. Refer to Figure 1 for the machine configuration. Many manufacturing environments that produce multiple final products are extensions of

---

\*Corresponding author. Email: [bmtlin@mail.nctu.edu.tw](mailto:bmtlin@mail.nctu.edu.tw)

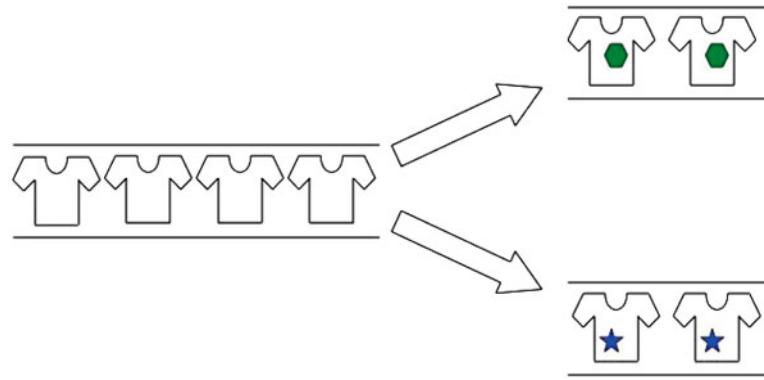


Figure 1. Differentiation flow shop model.

this basic model. We study the manufacturing environment shown in Figure 1. In such an environment, the stage-1 machine is common for all products. This means that the model can be used to achieve mass production of similar products in the first stage, and the products proceed to the stage-2 machines for further differentiation operations. As batch processing is common in mass production, we consider the production environment where batching is required on the stage-1 machine.

The rest of the paper is organised as follows. In Section 2, we formally define the studied problem and introduce the notation that will be used throughout the paper. A review of related works follows. As the studied problem is known to be NP-hard, we will study the scenario where the production sequence of products on each stage-2 machine is known and given. Section 3 presents an algorithm for the differentiation flow shop problem with batching. This algorithm is then applied to compute lower bounds on the optimal solutions of the original problem. The techniques are elaborated in Section 4. We give concluding remarks and suggest potential research directions in Section 5.

## 2. Problem statement and literature review

In this section, we first introduce the notation used in this paper. A formal definition of the studied problem follows. A numerical example will be given for illustration. Related works will also be reviewed.

### 2.1 Problem definition and notation

The notation that will be used throughout this paper is defined as follows.

#### Notation

$n_1$	number of type-1 jobs
$n_2$	number of type-2 jobs
$n = n_1 + n_2$	total number of all jobs to process
$I = \{I_1, I_2, \dots, I_{n_1}\}$	set of type-1 jobs
$J = \{J_1, J_2, \dots, J_{n_2}\}$	set of type-2 jobs
$M_0$	stage-1 common critical machine
$M_1$	stage-2 dedicated machine for jobs of $I$
$M_2$	stage-2 dedicated machine for jobs of $J$
$I_{i,m}$	operation of job $I_i \in I$ on stage $m$ ( $m = 1, 2$ )
$J_{j,m}$	operation of job $J_j \in J$ on stage $m$ ( $m = 1, 2$ )
$p_{I_i,m}$	processing time on stage $m$ , $m = 1, 2$ , of job $I_i \in I$
$p_{J_j,m}$	processing time on stage $m$ ( $m = 1, 2$ ) of job $J_j \in J$
$p_{(i),m}$	$i$ th shortest processing time on stage $m$ ( $m = 1, 2$ ) in $\{p_{I_1,m}, \dots, p_{I_{n_1},m}\}$
$p_{(j),m}$	$j$ th longest processing time on stage $m$ ( $m = 1, 2$ ) in $\{p_{J_1,m}, \dots, p_{J_{n_2},m}\}$
$s$	batch setup time on machine $M_0$
$B_l$	$l$ th batch on machine $M_0$

$C_{\max}^*(\mathcal{I}' \cup \mathcal{J}')$	optimal makespan of job set $\mathcal{I}' \cup \mathcal{J}'$ for $\mathcal{I}' \subseteq I$ and $\mathcal{J}' \subseteq J$
$S(\mathcal{I}' \cup \mathcal{J}')$	a particular schedule of job set $\mathcal{I}' \cup \mathcal{J}'$ for $\mathcal{I}' \subseteq I$ and $\mathcal{J}' \subseteq J$
$C_{\max}(S(\mathcal{I}' \cup \mathcal{J}'))$	makespan of schedule $S(\mathcal{I}' \cup \mathcal{J}')$
$S^*(\mathcal{I}' \cup \mathcal{J}')$	optimal schedule of job set $\mathcal{I}' \cup \mathcal{J}'$ for $\mathcal{I}' \subseteq I$ and $\mathcal{J}' \subseteq J$ , i.e. $C_{\max}(S^*(\mathcal{I}' \cup \mathcal{J}')) = C_{\max}^*(\mathcal{I}' \cup \mathcal{J}')$

The problem is formally defined as follows. The manufacturing model is a two-stage three-machine differentiation flow shop consisting of a stage-1 common critical machine and two independent dedicated machines in the second stage. The jobs belong to two different types: type 1,  $I = \{I_1, I_2, \dots, I_{n_1}\}$ , and type 2,  $J = \{J_1, J_2, \dots, J_{n_2}\}$ . There are in all  $n = n_1 + n_2$  jobs to process in the differentiation flow shop. Each job in  $I$  comprises two operations, the first of which is performed on the stage-1 common machine  $M_0$ , and the second is performed on the first dedicated machine  $M_1$ , as in the classical two-machine flow shop. Similarly, the jobs of  $J$  are processed first on the common critical machine and then on the second dedicated machine  $M_2$ . Jobs of both types are processed on the common critical machine in batches. A constant setup time  $s$  is required whenever a batch is formed on the stage-1 critical machine. The batch scheduling model we adopt in this paper is sum-batch or sequential-batch, under which the processing length of a batch is the setup time plus the sum of the processing times of the jobs contained in the batch. Batch availability is assumed, i.e. a job can proceed to the second stage only when all operations in the batch to which the job belongs are completely finished. We will investigate the compatible batch composition on the common machine, i.e. jobs from different types are allowed to be in the same batch. This assumption is due to the fact that the first machine is installed for mass production of generic products that will develop into different end-products through the stage-2 operations. The objective function considered is the makespan, i.e. the maximum completion time of all jobs. Denote the problem by  $F(1,2)|s\text{-batch}|C_{\max}$ , where  $F(1,2)$  indicates the flow shop and s-batch signifies the sequential-batch mode.

As the studied problem is known to be strongly NP-hard, this paper considers a simplified situation where the sequences of jobs of each type on the stage-1 machine are known *a priori* and fixed. The term ‘fixed\_seq’ is added to the problem notation to specify the restriction. Justification for the assumption of fixed sequences may be due to technical constraints or the first-come-first-served policy. Further justification stems from the need of calculating the objective value of a complete or partial sequence in the course of execution of branch-and-bound algorithms or meta-heuristics (Shafransky and Strusevich 1998; Cheng, Gupta, and Wang 2000; Lin and Hwang 2011; Hwang, Kovalyov, and Lin 2012). The latter studies investigate several interesting problems, for which determining optimal schedules from given sequences is not at all trivial. Subject to the assumption of given job sequences, the studied problem reduces to finding how to interleave two sequences of jobs and how to group the jobs in batches on the stage-1 machine. In addition, we consider only permutation schedules, that is, jobs of the same type have the same processing sequence on the critical machine and on their dedicated machines. Each machine can process at most one job at any time, and no preemption is allowed.

To illustrate the problem definition of  $F(1,2)|s\text{-batch}, \text{fixed\_seq}||C_{\max}$ , we give a numerical example. There are four jobs of two types to be scheduled: type 1,  $I = \{I_1, I_2\}$  and type 2,  $J = \{J_1, J_2\}$ . The batch setup time is 1. The processing times are as follows.

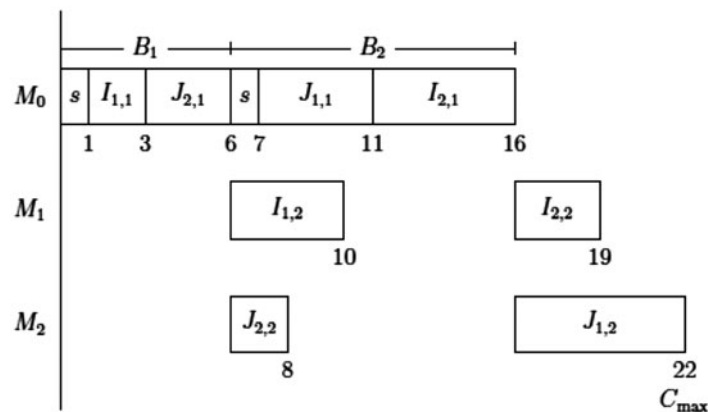
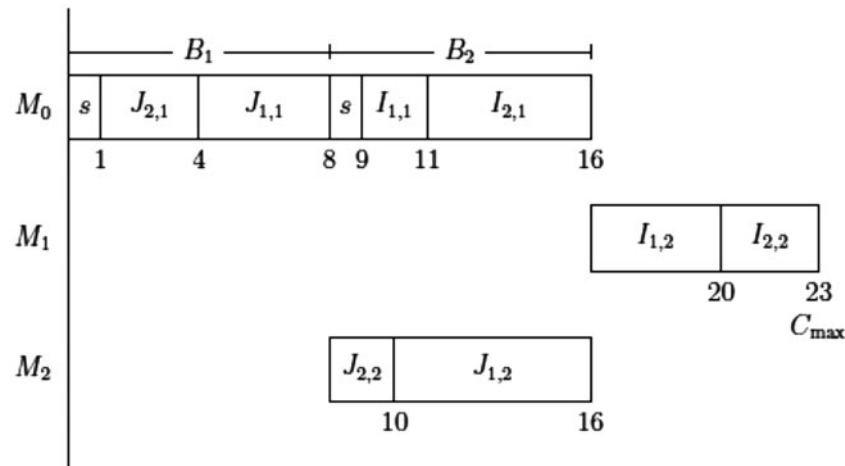


Figure 2. Gantt chart of batch sequence  $\sigma_1$ .

Figure 3. Gantt chart of batch sequence  $\sigma_2$ .

	Job			
	$I_1$	$I_2$	$J_1$	$J_2$
Stage 1	$p_{I_{1,1}} = 2$	$p_{I_{2,1}} = 5$	$p_{J_{1,1}} = 4$	$p_{J_{2,1}} = 3$
Stage 2	$p_{I_{1,2}} = 4$	$p_{I_{2,2}} = 3$	$p_{J_{1,2}} = 6$	$p_{J_{2,2}} = 2$

Given two batch sequences  $\sigma_1 = ((I_1, J_2), (J_1, I_2))$  and  $\sigma_2 = ((J_2, J_1), (I_1, I_2))$ , we have two corresponding Gantt charts as shown in Figures 2 and 3. Batch sequence  $\sigma_1$  has a makespan of 22 and batch sequence  $\sigma_2$  has a makespan of 23.

## 2.2 Literature review

It can readily be seen that when there is only one type of job and there is only one dedicated machine at stage 2, the problem becomes the classical two-machine flow shop scheduling problem, which can be solved in  $O(n \log n)$  time (Johnson 1954). The model of delayed differentiation studied in this paper was probably first investigated by Herrmann and Lee (1992), who investigate three objective functions, namely the makespan, the number of tardy jobs and the maximum lateness. They define two types of dispatching rules, look-ahead and look-behind, in their job shop scheduling approach. Look-ahead and look-behind scheduling include procedures that look around the shop for more information to use in making a scheduling decision. Look-ahead models consider the machines where the jobs will be headed after the present stage. Look-behind models, on the other hand, consider the job that will be arriving soon at the present machine. They use the terms look-ahead and look-behind to designate scheduling procedures that do more than consider just the state of one machine.

Drobouchevitch and Strusevich (2000) study the two-stage job shop scheduling problem with a bottleneck machine that can be thought of as a general case of our problem. Given an arbitrary number of stage-two machines in a job shop, they designed a heuristic algorithm for makespan minimisation with a performance ratio of 3/2. Without knowing the existence of the studies of Herrmann and Lee (1992) and Drobouchevitch and Strusevich (2000), Kyparisis and Koulamas (2000) investigate the same model but with  $m$  types of jobs, and correspondingly  $m$  dedicated machines at the second stage in the case of a flow shop and an open shop. Their model is polynomially solvable under the strong assumption, called the block assumption, that jobs of the same type must be processed contiguously on the stage-1 machine. Under this assumption, they develop a makespan minimisation algorithm in  $O(m(n \log n + \log m))$  time, where  $n$  is the total number of jobs of all types. Mosheiov and Yovel (2004) improve the algorithm of Kyparisis and Koulamas (2000) and reduce the time complexity to  $O(n \log n)$  subject to the common constraint  $m \leq n$ . The reverse model of delayed differentiation with dedicated machines installed at stage 1 was studied by Oğuz, Lin, and Cheng (1997), who show that minimising the makespan is ordinarily NP-hard by a reduction from Partition. A proof of strong NP-hardness is given by Lin (1999). Neumytov and Sevastyanov (1993) independently study the same problem focusing on the

approximation algorithms with their performance analysis. With the objective function of minimising the total machine completion time, Cheng, Lin, and Tian (2009) prove the problem to be strongly NP-hard, propose a heuristic and analyse its performance ratio. While the  $F(1,2)||C_{\max}$  model has been investigated in some studies, it is interesting to introduce the concept of batch processing to the stage-1 machine as it processes the operations common to all types of products. Batch scheduling has received considerable research attention in the past two decades, and there is a large body of research on this subject. The reader is referred to Cheng et al. (2000) and Potts and Kovalyov (2000), which are two excellent reviews providing a broad coverage and a comprehensive classification scheme.

Incorporating batch scheduling into the  $F(1,2)$  model is a practical generalisation, as the stage-1 machine processes the operations common to all product types. Batch scheduling has received considerable research attention in the past few decades. Relevant to our study, Ching, Liao, and Wu (1997) consider a two-stage setting similar to  $F(1,2)$  where each of the two product types has a total work load that can be partitioned into batches. A setup time is required when the processing of the stage-1 machine switches from one product type to the other. To minimise the makespan, they present an integer programming model and a heuristic algorithm without addressing the complexity issue. For the same model, Cheng and Kovalyov (1998) propose an  $O((n_1 + n_2)n_1^4n_2^3)$  dynamic programming algorithm, where  $n_1$  and  $n_2$  are the numbers of jobs of type 1 and type 2, for makespan minimisation. The algorithm is polynomial in the number of jobs, but not polynomial in input size because only seven parameters are involved. Considering the general setting of  $m$  product types, Cheng, Kovalyov, and Chakhlevich (2004) design an  $O(n^m)$  dynamic program. A recent paper by Gerstl and Mosheiov (2012) considers the three objectives of makespan, total machine completion time, and total weighted completion time for the case with two product types. Dynamic programming algorithms are proposed for the three objective functions. Similarly, the algorithms are not polynomial in input size. In the problems studied above, all follow the assumption that jobs of the same type are identical. The major distinct feature of the model studied in this paper is the relaxation that the jobs have arbitrary processing times.

### 3. Batching under two job sequences

This section is dedicated to the study of the batch scheduling problem with compatible batches to minimise the makespan. We will solve the problem subject to the assumption that the processing sequences of either type of job are known and fixed, i.e.  $F(1,2)|s\text{-batch, fixed\_seq}||C_{\max}$ . Given a batch scheduling system and the set of jobs to be processed, the scheduling problem is to determine the composition of each batch, i.e. the assignment of jobs to batches, the sequence of the batches to be processed, and the sequence of the jobs in each batch. The assumption of fixed sequences refers to the assumption that the job sequences of  $I$  and  $J$  are given and fixed. Since we have the assumption of fixed sequences, the decision remains as to how to interleave two given job sequences and also how to group the jobs into batches on the stage-1 machine. It is easy to show that there exists at least one optimal solution that is a permutation schedule, i.e. the processing sequences on both critical and dedicated machines of the two types of jobs are the same. Therefore, we consider only permutation schedules in this paper.

The first issue concerns optimally interleaving two sub-sequences without batching. Herrmann and Lee (1992) propose a polynomial-time algorithm to resolve this issue. Their approach first associates every job with a due-date and transforms the problem into the single-machine scheduling problem of minimising the maximum lateness,  $1||L_{\max}$ . Denote the job sequence of  $I$  as  $\sigma_I = (I_1, I_2, \dots, I_{n_1})$ , and the job sequence of  $J$  as  $\sigma_J = (J_1, J_2, \dots, J_{n_2})$ . The following algorithm can yield an interleaving sequence with the minimum makespan.

**Algorithm HL** (Herrmann and Lee 1992)

**Input:** Two job sequences  $\sigma_I$  and  $\sigma_J$

**Output:** A sequence of jobs of  $I \cup J$  with the minimum makespan

**Step 1:** For each job  $I_k$  of  $I$ , define  $A_k$  as the set of jobs (not including  $I_k$ ) that follow  $I_k$  in  $\sigma_I$ . Set due-date  $d_k = D - (p_{I_k,2} + \sum_{I_h \in A_k} p_{I_h,2})$ , where  $D$  is an arbitrarily large positive number

**Step 2:** For each job  $J_k$  of  $J$ , define  $A_k$  as the set of jobs (not including  $J_k$ ) that follow  $J_k$  in  $\sigma_J$ . Set due-date  $d_k = D - (p_{J_k,2} + \sum_{J_h \in A_k} p_{J_h,2})$

**Step 3:** Schedule the jobs on machine  $M_0$  in non-decreasing order of due-dates  $d_k$

Steps 1 and 2 of Algorithm HL respectively require  $O(n_1)$  and  $O(n_2)$  time to compute the due-dates. Step 3 takes  $O(n_1 + n_2) = O(n)$  time to interleave the two job sequences, since the jobs in each sequence are already sorted in non-increasing order of  $d_k$ , and forming the single job sequence on machine  $M_0$  is only to combine the two sequences without changing the relative ordering in each sequence. The question arises of whether or not the sequence produced



by Algorithm HL can serve as a basis for resolving  $F(1,2)|s\text{-batch}, \text{fixed\_seq}||C_{\max}$ , in which the operations on machine  $M_0$  are processed in batches. The following lemma confirms the validity of the algorithm for this specific job sequence.

**Lemma 3.1:** *There is an optimal schedule of  $F(1,2)|s\text{-batch}, \text{fixed\_seq}||C_{\max}$  in which the job sequence on machine  $M_0$  coincides with that reported by Algorithm HL.*

**Proof:** Assume that, in some optimal job sequence  $\sigma$ , there are two jobs arranged consecutively on machine  $M_0$  that are not sequenced as per Algorithm HL. The two jobs must be of different types. Denote them by  $I_i$  and  $J_j$ . Assume that  $d_{I_i} < d_{J_j}$  and that  $J_j$  precedes  $I_i$  in the optimal schedule. If  $I_i$  and  $J_j$  are in the same batch, we can swap their positions without affecting the job completion times in the schedule. If  $I_i$  and  $J_j$  are not in the same batch, then  $J_j$  is the last job in its batch and  $I_i$  is the first job in the succeeding batch. We move  $J_j$  into the succeeding batch. Two cases are analysed. If idle time exists after the completion of job  $J_j$  on machine  $M_2$ , then the completion time of machine  $M_2$  remains unchanged. If there is no idle time in the processing of job  $J_j$  and the remaining jobs on machine  $M_2$ , then by the inequality  $d_{I_i} < d_{J_j}$  (or  $p_{I_{i,2}} + \sum_{I_h \in A_k} p_{I_{h,2}} > p_{J_{j,2}} + \sum_{J_h \in A_k} p_{J_{h,2}}$ ) the completion time of machine  $M_2$  is not greater than that of machine  $M_1$ . Therefore, arranging job  $I_i$  in front of job  $J_j$  will not increase the makespan. Repeating the above argument, if necessary, we will come up with a job sequence as in Algorithm HL.  $\square$

While the interleaving issue is resolved, we proceed to the batching issue. On a single machine, problem  $1|s\text{-batch}|L_{\max}$  is to sequence as well as batch the jobs so as to minimise the maximum lateness.

**Lemma 3.2:** (Webster and Baker 1995) *There is an optimal schedule for the  $1|s\text{-batch}|L_{\max}$  problem in which the jobs are sequenced by the earliest due-date first (EDD) rule.*

Based upon Theorem 3.3, Webster and Baker (1995) develop an  $O(n^2)$  backward dynamic programming algorithm for the  $1|s\text{-batch}|L_{\max}$  problem.

**Theorem 3.3:** (Webster and Baker 1995) *The  $1|s\text{-batch}|L_{\max}$  problem can be solved in  $O(n^2)$  time.*

In addition to the above known results, we can solve the  $F(1,2)|s\text{-batch}, \text{fixed\_seq}||C_{\max}$  problem optimally in polynomial time as follows. We first treat the operations of jobs of  $I \cup J$  on machine  $M_0$  as jobs and associate them with due-dates as defined in Algorithm HL. Step 3 of Algorithm HL produces an EDD job sequence on machine  $M_0$ . Then we apply Algorithm WB to optimally group the jobs in the sequence.

**Theorem 3.4:** *Problem  $F(1,2)|s\text{-batch}, \text{fixed\_seq}||C_{\max}$  can be solved in  $O(n^2)$  time.*

#### 4. Lower bound

This section returns to the general  $F(1,2)|s\text{-batch}|C_{\max}$  problem without the assumption of two fixed sequences. Since  $F(1,2)||C_{\max}$  was proven to be strongly NP-hard by a reduction from a 3-partition by Herrmann and Lee (1992), problem  $F(1,2)|s\text{-batch}|C_{\max}$  with batching considerations is also hard to solve. The NP-hardness indicates that it is very unlikely to design a polynomial-time algorithm for producing optimal solutions. Effective lower bounds are crucial to the efficiency of branch-and-bound algorithms and can be used to measure the effectiveness of approximation approaches. We will develop a lower bound using the result of Theorem 3.4 and the techniques of Cheng, Lin, and Toker (2000) and Lin and Wu (2005).

For any job set  $I \cup J$  to be processed, we derive another job set  $\mathcal{I}' \cup \mathcal{J}'$  by rearranging the operations of the jobs to create an *ideal* data set. In makespan minimisation of a two-machine flow shop problem, a job is preferred to be processed first if its processing time on stage 1 is shorter and its processing time on stage 2 is longer. A data set is ideal if and only if the data set contains jobs where a job with a shorter processing time on stage 1 has a longer processing time on stage 2. In the following, we create such an ideal data set  $\mathcal{I}' \cup \mathcal{J}'$  from the given job set  $I \cup J$ . For all  $k = 1, 2, \dots, n_1$ , job  $\mathcal{I}'_k$  of set  $\mathcal{I}'$  is defined by two parameters:

- (1)  $p_{\mathcal{I}'_k,1} = p_{I_{(k),1}}$ , i.e. the  $k$ th smallest element among  $p_{I_{1,1}}, p_{I_{2,1}}, \dots, p_{I_{n_1,1}}$ ; and
- (2)  $p_{\mathcal{I}'_k,2} = p_{I_{(k),2}}$ , i.e. the  $k$ th largest element among  $p_{I_{1,2}}, p_{I_{2,2}}, \dots, p_{I_{n_1,2}}$ .

In a similar way, job  $\mathcal{J}'_k$  of  $\mathcal{J}'$  is defined by  $p_{J_{(k),1}}$  and  $p_{J_{(k),2}}$  for all  $k = 1, 2, \dots, n_2$ . We index the jobs in  $\mathcal{I}'$  and  $\mathcal{J}'$  in non-decreasing order of their processing times on machine  $M_0$  and denote the sequence of set  $\mathcal{I}'$  (respectively, set  $\mathcal{J}'$ ) as  $\sigma_{\mathcal{I}'} = (\mathcal{I}'_1, \mathcal{I}'_2, \dots, \mathcal{I}'_{n_1})$  (respectively,  $\sigma_{\mathcal{J}'} = (\mathcal{J}'_1, \mathcal{J}'_2, \dots, \mathcal{J}'_{n_2})$ ).

**Lemma 4.1:** *There is an optimal schedule of the instance  $\mathcal{I}' \cup \mathcal{J}'$  where jobs of  $\mathcal{I}'$  are sequenced as in  $\sigma_{\mathcal{I}'}$  and jobs of  $\mathcal{J}'$  are sequenced as in  $\sigma_{\mathcal{J}'}$ .*

**Proof:** Without loss of generality, we assume that the processing times of all operations are distinct. Given a schedule of the instance  $S(\mathcal{I}' \cup \mathcal{J}')$ , for each pair of jobs  $\{(\mathcal{I}'_k, \mathcal{I}'_{k+1}) | k = 1, 2, \dots, n_1 - 1\}$  in  $S(\mathcal{I}' \cup \mathcal{J}')$ , we swap their position if  $p_{\mathcal{I}'_{k,1}} > p_{\mathcal{I}'_{k+1,1}}$ . Note that, if the condition is met, it implies  $p_{\mathcal{I}'_{k,2}} < p_{\mathcal{I}'_{k+1,2}}$ . A similar job-interchange technique is applied to job set  $\mathcal{J}'$ . Since the total idle time on either dedicated machine will not increase after the job interchange, it is clear that the makespan will not increase in the derived schedule. Repeating the job interchange, if necessary, will finally lead to a schedule in which all jobs of  $\mathcal{I}'$  are sequenced by  $\sigma_{\mathcal{I}'}$  and jobs of  $\mathcal{J}'$  are sequenced by  $\sigma_{\mathcal{J}'}$ .  $\square$

**Lemma 4.2:** *An optimal schedule of the job set  $\mathcal{I}' \cup \mathcal{J}'$  can be found in  $O(n^2)$  time.*

**Proof:** To decide how to group jobs into batches, it can readily be done using Algorithm WB, which has the time complexity  $O(n^2)$ . Therefore, an optimal schedule of the job set  $\mathcal{I}' \cup \mathcal{J}'$  can be found in  $O(n^2)$  time and the lemma follows.  $\square$

**Lemma 4.3:** *The optimal makespan of the job set  $\mathcal{I}' \cup \mathcal{J}'$  will be no greater than the optimal makespan of the original job set  $I \cup J$ , i.e.  $C_{\max}^*(\mathcal{I}' \cup \mathcal{J}') \leq C_{\max}^*(I \cup J)$ .*

**Proof:** Consider the schedule  $S^*(I \cup J)$ , i.e. the schedule of job set  $I \cup J$  with the optimal makespan or  $C_{\max}^*(I \cup J)$ . For each pair of the stage-1 (respectively, stage-2) operations of jobs  $\{(I_k, I_{k+1}) | k = 1, 2, \dots, n_1 - 1\}$ , we swap their positions if  $p_{I_{k,1}} > p_{I_{k+1,1}}$  (respectively,  $p_{I_{k,2}} < p_{I_{k+1,2}}$ ) and leave their stage-2 (respectively, stage-1) operations unaltered in their original positions. In a similar way, we apply the operation-interchange technique to the operations of  $J$  on both stage-1 and stage-2 operations. Clearly, the makespan will not increase after we change the positions of the operations under the specific condition. Therefore, the derived schedule contains jobs of the job set  $\mathcal{I}' \cup \mathcal{J}'$  and has a makespan that is no worse than the previous makespan in the same batch composition, i.e.  $C_{\max}(S(\mathcal{I}' \cup \mathcal{J}')) \leq C_{\max}^*(I \cup J)$ . By Lemma 4.1, the optimal makespan of job set  $\mathcal{I}' \cup \mathcal{J}'$  is minimum of those of all possible schedules of job set  $\mathcal{I}' \cup \mathcal{J}'$  or  $C_{\max}^*(\mathcal{I}' \cup \mathcal{J}') \leq C_{\max}(S(\mathcal{I}' \cup \mathcal{J}'))$ . By transitivity, we can see that  $C_{\max}^*(\mathcal{I}' \cup \mathcal{J}') \leq C_{\max}^*(I \cup J)$  and complete the proof.  $\square$

In summary, by rearranging the operations in the two given job sets, we obtain two data sets,  $\mathcal{I}'$  and  $\mathcal{J}'$ , which constitute an instance that exhibits an ideal structure and permits fast solution algorithms. With the two fixed sequences of jobs, the scheduling problem is now equivalent to  $F(1,2)|s\text{-batch, fixed\_seq}|C_{\max}$ , which can be solved optimally using the algorithm in the previous section (by Theorem 3.4). In addition, we obtain the minimum makespan of the derived instance, which serves as a lower bound for the original problem.

**Theorem 4.4:** *A lower bound of  $F(1,2)|s\text{-batch}|C_{\max}$  can be found in  $O(n^2)$ .*

**Proof:** Constructing sets  $\mathcal{I}'$  and  $\mathcal{J}'$  takes  $O(n_1 \log n_1) + O(n_2 \log n_2) = O(n \log n)$  time because of the sorting operations. Interleaving two sequences takes  $O(n)$  time, and Algorithm WB takes  $O(n^2)$  time to group the jobs. Therefore, the overall time complexity to obtain a lower bound of  $F(1,2)|s\text{-batch}|C_{\max}$  is  $O(n^2)$ .  $\square$

An example is given below to illustrate the lower bound calculation for problem  $F(1,2)|s\text{-batch}|C_{\max}$ . There are four jobs of two types to be scheduled: type 1,  $I = \{I_1, I_2\}$ , and type 2,  $J = \{J_1, J_2\}$ . The batch setup time is 1. The processing times of jobs are as follows.



	Job			
	$I_1$	$I_2$	$J_1$	$J_2$
Stage 1	$p_{I_1,1} = 2$	$p_{I_2,1} = 7$	$p_{J_1,1} = 3$	$p_{J_2,1} = 4$
Stage 2	$p_{I_1,2} = 3$	$p_{I_2,2} = 9$	$p_{J_1,2} = 2$	$p_{J_2,2} = 6$

An ideal job set  $\mathcal{I}' \cup \mathcal{J}'$  can be derived from the given job set  $I \cup J$  by the operation interchange technique. The processing times of jobs in  $\mathcal{I}' \cup \mathcal{J}'$  are as follows.

	Job			
	$I_{1'}$	$I_{2'}$	$J_{1'}$	$J_{2'}$
Stage 1	$p_{I_{1'},1} = 2$	$p_{I_{2'},1} = 7$	$p_{J_{1'},1} = 3$	$p_{J_{2'},1} = 4$
Stage 2	$p_{I_{1'},2} = 9$	$p_{I_{2'},2} = 3$	$p_{J_{1'},2} = 6$	$p_{J_{2'},2} = 2$

Applying Algorithm HL, we can determine the sequence of jobs on machine  $M_0$ . Let  $D = 0$ . We can obtain the due-dates of all jobs as follows.

	Job			
	$I_{1'}$	$J_{1'}$	$I_{2'}$	$J_{2'}$
Stage 1	$p_{I_{1'},1} = 2$	$p_{J_{1'},1} = 3$	$p_{I_{2'},1} = 7$	$p_{J_{2'},1} = 4$
Stage 2	$p_{I_{1'},2} = 9$	$p_{J_{1'},2} = 6$	$p_{I_{2'},2} = 3$	$p_{J_{2'},2} = 2$
$d_k$	$d_{I_{1'}} = 12$	$d_{J_{1'}} = 8$	$d_{I_{2'}} = 3$	$d_{J_{2'}} = 2$
$D_k$	$D_{I_{1'}} = -12$	$D_{J_{1'}} = -8$	$D_{I_{2'}} = -3$	$D_{J_{2'}} = -2$

The following is the recursive steps of Algorithm WB:

$$B(1) = \min \begin{cases} 1 + \sum_{k=1}^1 p_k + \max\{12, B(2)\} \\ 1 + \sum_{k=1}^2 p_k + \max\{12, B(3)\} \\ 1 + \sum_{k=1}^3 p_k + \max\{12, B(4)\} \\ 1 + \sum_{k=1}^4 p_k + \max\{12, B(5)\} \end{cases}$$

$$B(2) = \min \begin{cases} 1 + \sum_{k=2}^2 p_k + \max\{8, B(3)\} \\ 1 + \sum_{k=2}^3 p_k + \max\{8, B(4)\} \\ 1 + \sum_{k=2}^4 p_k + \max\{8, B(5)\} \end{cases}$$

$$B(3) = \min \begin{cases} 1 + \sum_{k=3}^3 p_k + \max\{3, B(4)\} \\ 1 + \sum_{k=3}^4 p_k + \max\{3, B(5)\} \end{cases}$$

$$B(4) = \min\{1 + \sum_{k=4}^4 p_k + \max\{2, B(5)\}\}$$

$$B(5) = -\infty$$

$$B(4) = \min\{1 + 4 + \max\{2, -\infty\}\} = 7$$

$$B(3) = \min \left\{ \begin{array}{l} 1 + 7 + \max\{3, 7\} \\ 1 + 11 + \max\{3, -\infty\} \end{array} \right\} = 15$$

$$B(2) = \min \left\{ \begin{array}{l} 1 + 3 + \max\{8, 15\} \\ 1 + 10 + \max\{8, 7\} \\ 1 + 14 + \max\{8, -\infty\} \end{array} \right\} = 19$$

$$B(1) = \min \left\{ \begin{array}{l} 1 + 2 + \max\{12, 19\} \\ 1 + 5 + \max\{12, 15\} \\ 1 + 12 + \max\{12, 7\} \\ 1 + 16 + \max\{12, -\infty\} \end{array} \right\} = 22.$$

The optimal makespan of job set  $\mathcal{I}' \cup \mathcal{J}'$  is 22, and the corresponding schedule is  $((I_1', J_1'), (I_2'), (J_2'))$ . The value 21 is a lower bound on the optimal makespan of job set  $I \cup J$ .

## 5. Concluding remarks

This paper investigates a makespan minimisation scheduling problem in a two-stage three-machine flow shop, known as a differentiation flow shop that has a critical machine at stage 1 and two independent dedicated machines at stage 2. The new feature incorporated in this work is that the first-stage machine processes the jobs in batches subject to the assumption of continuous processing and batch availability. The studied problem is computationally intractable. We solve a special case, in which two given fixed sequences of two types of jobs are given, by developing a polynomial-time algorithm. The algorithm is deployed to derive a lower bound of the general case in  $O(n^2)$  time. The lower bound can be used for the design of branch-and-bound algorithms or the evaluation of heuristic approaches.

This study was focused on the development of properties for producing lower bounds. For further research, it would be interesting to conduct a computational study of the effectiveness of the derived bounds as well as the efficiency curtailung unnecessary branching in branch-and-bound algorithms. The development of heuristics along with a computational study would also be useful for appraisal of the gap between the lower bound and upper bounds. It may also be worthwhile generalising the studied problem to the setting with a variable number of parallel dedicated machines, i.e. the number  $m$  of parallel dedicated machines is part of the input. Even though this problem is clearly NP-hard in the strong sense, it may be interesting to determine the complexity status of the  $F(1, m)$  model in special cases such as (1) all jobs have the same processing time on the critical machine, or (2) sequences of different types of jobs are fixed and given.

## Acknowledgements

The authors are grateful to the anonymous reviewers for their constructive comments. This research was supported, in part, by the National Science Council of Taiwan under grant NSC98-2410-H-009-011-MY2.

## References

- Bruce, L. 1987. "The Bright New Worlds of Benetton." *International Management* 42: 24–35.
- Cheng, T. C. E., J. N. D. Gupta, and G. Wang. 2000. "A Review of Flowshop Scheduling Research with Setup Times, Production and Operations Management." *Production and Operations Management* 9 (3): 262–282.
- Cheng, T. C. E., and M. Y. Kovalyov. 1998. "An Exact Algorithm for Batching and Scheduling in a Mixed Shop." *International Journal of Production Economics* 55 (1): 53–56.
- Cheng, T. C. E., M. Y. Kovalyov, and K. N. Chakhlevich. 2004. "Batching in a Two-Stage Flowshop with Dedicated Machines in Second Stage." *IIE Transactions* 36 (1): 87–93.
- Cheng, T. C. E., B. M. T. Lin, and Y. Tian. 2009. "Minimizing the Weighted Sum of Machine Completion times in a Two-Stage Flow Shop with a Critical Machine." *Computers & Operations Research* 36: 3031–3040.
- Cheng, T. C. E., B. M. T. Lin, and A. Tokar. 2000. "Makespan Minimization in the Two-Machine Flowshop Batch Scheduling Problem." *Naval Research Logistics* 47 (2): 128–144.

- Ching, C. Y., C. J. Liao, and C. J. Wu. 1997. "Batching to Minimize Total Production Time for Two Part Types." *International Journal of Production Economics* 48 (1): 63–72.
- Da Silveira, G., D. Borenstein, and F. S. Fogliatto. 2001. "Mass Customization: Literature Review and Research Directions." *International Journal of Production Economics* 72 (1): 1–13.
- Drobouchevitch, I. G., and V. A. Strusevich. 2000. "Heuristics for the Two-Stage Job Shop Scheduling Problem with a Bottleneck Machine." *European Journal of Operational Research* 123 (2): 229–240.
- Dudek, R. A., S. S. Panwalkar, and M. L. Smith. 1992. "The Lessons of Flowshop Scheduling Research." *Operations Research* 40 (1): 7–13.
- Eppen, G. 1979. "Effects of Centralization on Expected Costs in Multi-Location Newsboy Problems." *Management Science* 25 (5): 498–501.
- Feitzinger, E., and H. Lee. 1997. "Mass Customization at Hewlett Packard: The Power of Postponement." *Harvard Business Review* 75: 116–121.
- Gerstl, E., and G. Mosheiov. 2012. "A Two-Stage Flow Shop Scheduling with a Critical Machine and Batch Availability." *Foundations of Computing and Decision Sciences* 37 (1): 39–56.
- Gupta, D., and S. Benjaafar. 2004. "Make-to-Order, Make-to-Stock, or Delay Product Differentiation? A Common Framework for Modeling and Analysis" *IIE Transactions* 36 (6): 529–546.
- Herrmann, J. W., and C. Y. Lee. 1992. *Three-Machine Look-Ahead Scheduling Problems*. Research Report. 92–93. University of Florida, 1993.
- Hwang, F. J., M. Y. Kovalyov, and B. M. T. Lin. 2012. "Total Completion Time Minimization in Two-Machine Flow Shop Scheduling Problems with a Fixed Job Sequence." *Discrete Optimization* 9 (1): 29–39.
- Johnson, S. M. 1954. "Optimal Two and Three-Stage Production Schedules with Setup Times Included." *Naval Research Logistics Quarterly* 1: 61–68.
- Kyparisis, G. J., and C. Koulamas. 2000. "Flow Shop and Open Shop Scheduling with a Critical Machine and Two Operations Per Job." *European Journal of Operational Research* 127 (1): 120–125.
- Lin, B. M. T. 1999. "The Strong NP-Hardness of Two-Stage Flowshop Scheduling Problem with a Common Second-Stage Machine." *Computers & Operations Research* 27 (6): 695–698.
- Lin, B. M. T., and F. J. Hwang. 2011. "Total Completion Time Minimization in a 2-Stage Differentiation Flowshop with Fixed Sequences per Job Type." *Information Processing Letters* 111 (5): 208–212.
- Lin, B. M. T., and J. M. Wu. 2005. "A Simple Lower Bound for Two Machine Flowshop Scheduling to Minimize Total Completion Time." *Asia Pacific Journal of Operational Research* 22 (3): 391–408.
- Magretta, J. 1998. "The Power of Virtual Integration: An Interview with Dell Computer's Michael Dell." *Harvard Business Review* 76: 72–84.
- Mosheiov, G., and U. Yovel. 2004. "Comments on Flow Shop and Open Shop Scheduling with a Critical Machine and Two Operations per Job." *European Journal of Operational Research* 157 (1): 257–261.
- Neumytov, Y. D., and S. V. Sevastyanov. 1993. "Approximation Algorithm with Tight Bound for Three-machine Counter-routes Problem." *Upravlyaemye Sistemy* 31: 53–65.
- Oğuz, C., B. M. T. Lin, and T. C. E. Cheng. 1997. "Two-Stage Flowshop Scheduling Problem with a Common Second-Stage Machine." *Computers & Operations Research* 24 (12): 1169–1174.
- Pinedo, M. 2012. *Scheduling: Theory, Algorithms and Systems*. New York: Springer.
- Pinedo, M., and X. Chao. 1999. *Operations Scheduling with Applications in Manufacturing and Services*. Boston: McGraw-Hill.
- Potts, C. N., and M. K. Kovalyov. 2000. "Scheduling with Batching: A Review." *European Journal of Operational Research* 102 (2): 228–249.
- Reisman, A., A. Kumar, and J. Motwani. 1997. "Flowshop Scheduling/Sequencing Research: A Statistical Review of the Literature, 1952–1994." *IEEE Transactions on Engineering Management* 44 (3): 316–329.
- Shafransky, Y. M., and V. A. Strusevich. 1998. "The Open Shop Scheduling Problem with a Given Sequence of Jobs on One Machine." *Naval Research Logistics* 45 (7): 705–731.
- Simchi-Levi, D., P. Kaminsky, and E. Simchi-Levi. 2007. *Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies*. Boston: McGraw-Hill.
- Webster, S., and K. R. Baker. 1995. "Scheduling Groups of Jobs on a Single Machine." *Operations Research* 43 (4): 692–703.