Operational Research in Engineering Sciences: Theory and Applications
Vol. 4, Issue 3, 2021, pp. 142-175
ISSN: 2620-1607
eISSN: 2620-1747
DOI: https://doi.org/10.31181/oresta111221142c

SCHEDULING WITH LOT STREAMING IN A TWO-MACHINE RE-ENTRANT FLOW SHOP

Ferda Can Çetinkaya [1]*, Mehmet Duman [2]

[1] Department of Industrial Engineering, Çankaya University, Ankara, Turkey
[2] NERITA, Near East University, TRNC, Mersin 10, Turkey

Received: 16 August 2021
Accepted: 24 November 2021
First online: 11 December 2021

Research paper

Abstract: *Lot streaming is splitting a job-lot of identical items into several sublots (portions of a lot) that can be moved to the next machines upon completion so that operations on successive machines can be overlapped; hence, the overall performance of a multi-stage manufacturing environment can be improved. In this study, we consider a scheduling problem with lot streaming in a two-machine re-entrant flow shop in which each job-lot is processed first on Machine 1, then goes to Machine 2 for its second operation before it returns to the primary machine (either Machine 1 or Machine 2) for the third operation. For the two cases of the primary machine, both single-job and multi-job cases are studied independently. Optimal and near-optimal solution procedures are developed. Our objective is to minimize the makespan, which is the maximum completion time of the sublots and job lots in the single-job and multi-job cases, respectively. We prove that the single-job problem is optimally solved in polynomial-time regardless of whether the third operation is performed on Machine 1 or Machine 2. The multi-job problem is also optimally solvable in polynomial time when the third operation is performed on Machine 2. However, we prove that the multi-job problem is NP-hard when the third operation is performed on Machine 1. A global lower bound on the makespan and a simple heuristic algorithm are developed. Our computational experiment results reveal that our proposed heuristic algorithm provides optimal or near-optimal solutions in a very short time.*

Key words: *scheduling, lot streaming, two-machine, re-entrant flow shop, makespan.*

1. Introduction

Manufacturing systems vary from a simple one-stage environment to more complex environments, such as a general job shop system, where jobs have different routings through multiple stages. Since the well-known efficient scheduling algorithm for the basic two-machine flow shop system, in which the flow of each job

* Corresponding author.
cetinkaya@cankaya.edu.tr (F. C. Çetinkaya), mehmet.duman@neu.edu.tr (M. Duman)

is the same, was proposed by Johnson (1954), scheduling problems in different and more complicated manufacturing environments have been extensively studied. The ***re-entrant flow shop***, which is a relatively new flow shop manufacturing environment, has drawn researchers' attention. In the re-entrant flow shop, a job has to re-visit some of the machines since the number of operations for each job is more than the number of machines (Lev & Adiri, 1984). We observe re-entrant Re-entrant flow shops can be observed mainly in textile and high-tech industries, such as printing printed circuit boards, wafer fabrications, and signal processing.

In all of the studies in the literature for the re-entrant flow shops and most of the scheduling studies for the other multi-stage manufacturing systems, it is assumed that jobs are indivisible entities. Thus, an operation of a ***job-lot*** (i.e., a process batch) consisting of identical units must be finished before it this job lot is transferred to the next machine. However, in many industrial applications, a job lot can be split into several ***sublots*** (i.e., transfer batches), which are the partial batches of the process batch. Transfer of the processed sublots to downstream machines without waiting for the completion of the whole job-lot on a machine gives an opportunity of allows the operations overlapping to overlap. The process of simultaneously splitting a job-lot into sublots and scheduling those sublots by overlapping their operations is known as ***lot streaming***, which was first mentioned as a scheduling technique by Reither (1966).

In this paper, we consider a scheduling problem with lot streaming in a two-machine re-entrant flow shop where each job-lot is processed first on Machine 1, then goes to Machine 2 for its second operation before it returns to the primary machine (either Machine 1 or Machine 2) for the third operation. We first focus on the single-job problem where a job-lot is spilt split into a given number of consistent or variable sublots. ***Consistent sublots*** case is the case where the size of each sublot does not change over the machines. However, the sublot sizes may change vary over the machines when ***variable sublots*** are used. Next, we extend the problem to the multi-job case in which the size of sublots and the schedule of multiple sublots and job lots need to be determined simultaneously. Our objective is to minimize the makespan, equivalent to the time to complete the last sublot in the single-job case, whereas it is the time to complete the last job lot in the multi-job case. Makespan aims to increase the utilization of the machines in the shop. To the best of our knowledge, our study is the only one that applies lot streaming for single- and multi-job cases in the re-entrant flow shops.

The organization of the remaining parts is as follows. The following section presents a literature review on lot streaming problems, especially those in two- and three-machine manufacturing shops. In Section 3, the single-job problem is studied, and the optimal schedules with consistent and variable sublots are developed. Section 4 considers the multi-job lot streaming problem and gives an exact algorithm that determines the optimal consistent-sublot sizes and job schedules when the primary machine is Machine 2. Next, the multi-job lot streaming problem, in which the primary machine is Machine 1, is proved to be strongly NP-hard, and a polynomial-time solvable case of the problem is provided. Moreover, a heuristic algorithm is provided for the multi-job problem where the primary machine is Machine 1, and its effectiveness is computationally tested. Finally, our brief conclusions and some issues for future research are summarized in Section 5.

2. Literature review

Although lot steaming is known and used in practice, there were no analytical studies in the literature of scheduling problems until the late '80s. Since then, lot streaming has attracted significant interest from researchers dealing with scheduling problems.

Based on the number of job lots, the studies in the literature can be divided into two categories. One category is called the ***single-job lot problem*** and deals with the sublot-sizing problem in which the sublot sizes are determined when there is a single job lot. The other category is called ***multi-job problem*** and deals with the ***sublot-sizing*** and ***job-sequencing*** subproblems simultaneously. Here, we limit our literature review to the lot streaming studies for the single- and multi-job lot cases in two- and three-machine manufacturing shops only to expose the proper place of our study in the literature.

The study by Baker (1988) is the first one considering the lot streaming technique for a single-job problem in two- or three-machine flow shops to minimize the makespan. For the two-machine case, he provided a linear programming model and determined the sublot sizes optimally. He also determined the optimal sublot sizes for the three-machine case where the job-lot is split into two sublots only. Potts & Baker (1989) proved that optimal sublots are consistent in the two-machine flow shop and illustrated that the consistent sublots are not always optimal for the flow shops with three and more machines. Glass et al. (1994) developed an algorithm that determines the optimal consistent-sublot sizes for the three-machine flow shop to minimize the makespan. This study was extended to the cases with sequence-independent detached and attached setups by Chen & Steiner (1997) and Chen & Steiner (1998), respectively. In the ***attached setup*** case of a machine, the first sublot belonging to a job lot should be available before setting up this machine. However, in the ***detached setup*** case, no need to wait for the arrival of the job lot. In both cases, no setup is necessary between successive sublots of the same job lot. In both cases, no setup is necessary between successive sublots of the same job lot. On the other hand, variable sublots case of the single job-lot problem was first examined by Trietsch (1989). The optimal solution with variable sublots in the three-machine flow shop was proposed by Trietsch & Baker (1993). Alfieri et al. (2012) and Alfieri et al. (2021) proposed exact, and heuristic solution approaches based on dynamic programming for a single-job problem to minimize the makespan and total flow time, respectively, in a two-machine flow shop with attached setup times.

Vickson & Alfredson (1992) considered the concept of lot streaming for scheduling multiple job lots in flow shops. They demonstrated that a modified Johnson's algorithm with unit-sized sublots solves the two-machine makespan minimization problem when the number of sublots in each jot-lot is unlimited and proved that sublots in each job-lot should be processed successively without the intermingling of different job lots. Vickson & Alfredson's study was extended by Çetinkaya & Kayalıgil (1992) to develop a unified algorithm that treats sequence-independent attached and detached setups. Çetinkaya (1994) considered the scheduling of multiple job lots in a two-machine flow shop with attached setup and removal times on the machines and proved that the optimal schedule of the job lots to minimize the makespan is obtained by determining the equal or unequal sublots sizes of each job-lot independently and sequencing the job lots by a modified

Johnson's algorithm. Vickson (1995) provided an optimal solution for the multi-job problem in a two-machine flow shop with sequence-independent attached or detached setups and transfer times from the first machine to the second one. Pranzo (2004) extended Çetinkaya's study by considering limited buffers between machines. Glass & Possani (2011) considered the two-machine flow shop problem with attached setup and transportation times to minimize the makespan of the multiple job lots. Their study showed that sublot-sizing and job-sequencing problems are solved independently, as in Çetinkaya (1994) and Vickson (1995), and provided an algorithm solvable in polynomial time. Baker (1995) considered the multi-job problem with equal-sized sublots and setup times in a two-machine flow shop and proposed an algorithm using the time-lag approach. Yang & Chern (2000) extended Baker's study to where detached setup times, transportation times, and removal times exist. Sriskandarajah & Wagneur (1999) investigated the multi-job problem in a two-machine flow shop with no-wait constraint. Çetinkaya (2005) considered a two-machine flow shop with a single agent transferring a completed item from Machine 1 to Machine 2. Machine 1 is blocked while the transport agent is in transferring and returning. He provided an algorithm that determines the optimal schedule for the case with unit-sized sublots.

From the early 2000s, researchers began to consider flow shops with more than three machines (stages). Several metaheuristic algorithms, such as genetic algorithms (Yoon & Ventura, 2002; Marimuthu et al., 2008; Martin, 2009; Defersha & Chen, 2010), discrete particle swarm optimization algorithm (Tseng & Liao, 2008), threshold accepting and ant-colony optimization algorithms (Marimuthu et al., 2009), discrete artificial bee colony algorithm (Pan et al., 2011), and migrating birds optimization algorithm (Devendra et al., 2014; Meng et al., 2018) have been proposed to solve the multi-job lot streaming problem by considering its different aspects.

All the studies mentioned above for the multi-job scheduling with lot streaming in the flow shops with more than three machines assume that sublots in each job lot should be processed successively for each operation on each machine. i.e., the intermingling of the job lots is not allowed, and the schedules are permutation schedules where the sequence of job lots on all machines is the same. However, a more realistic case is when intermingling of the job lots and non-permutation schedules are allowed. Feldmann & Biskup (2008) investigated the permutation flowshop scheduling problem with lot streaming and intermingling and developed a mixed-integer programming model. Rossit et al. (2016) investigated this non-permutation flowshop scheduling problem with lot streaming. They proposed a mathematical model to minimize the makespan of the multiple job lots that are not allowed to be intermingled.

Besides the studies mentioned above, lot streaming studies for other two- and three-machine manufacturing shops are scarce. There are studies considering open shops (Şen & Benli, 1999), hybrid flow shops (Kim et al., 1997; Zhang et al., 2003; Zhang et al., 2005; Liu, 2008; Defersha, 2011; Defersha & Chen, 2012a; Naderi & Yazdani, 2015; Cheng et al., 2016; Zhang et al., 2017; Wang et al., 2019; Li et al., 2020), and mixed shops (Çetinkaya & Duman, 2010), job shops (Buscher & Shen, 2009; Defersha & Chen, 2012b), and assembly shops (Sarin et al., 2011; Yao & Sarin, 2014; Nejati et al., 2016; Cheng & Sarin, 2020).

The comprehensive surveys by Chang & Chiu (2005), Sarin & Jaiprakash (2007), Gomez-Gasquet et al. (2013), and Cheng et al. (2013), and Salazar-Moya & Garcia (2021) are also available for lot streaming problems with job lots having more than one operation in multiple machines environments.

As we can see from the lot streaming literature and to the best of our knowledge, there is no previous study dealing with lot streaming for single- and multi-job cases in the re-entrant flow shops. The main contributions of our study can be summarized as follows:

- This study is the first one in the scheduling literature dealing with lot streaming in the re-entrant flow shops.
- Our study proves that the single-job problem is polynomial-time solvable regardless of whether the third operation is performed on Machine 1 or Machine 2 and develops optimal schedules with closed formulae for the optimal consistent and variable sublot sizes.
- Our study also proves that the multi-job problem is polynomial-time solvable when the third operation is performed on Machine 2 and develops optimal schedules with closed formulae for the optimal sublot sizes. However, the multi-job problem is *NP*-hard when the third operation is performed on Machine 1.
- A global lower bound on the makespan and a simple heuristic algorithm providing optimal or near-optimal schedules have been developed.

3. Single-job case

Our single-job problem in the two-machine re-entrant flow shop is explained as follows: A job-lot of U identical items has three operations to be performed. Each operation k $(k = 1,2,3)$ requires p_k time units of processing. There are two machines M_1 (Machine 1) and M_2 (Machine 2) operating independently. The first and second operations of the job-lot are performed on M_1 and M_2, respectively. A ***primary machine*** (M_1 or M_2) is re-visited by each item of the job lot for its third operation; hence the shop is a re-entrant flow shop. The job-lot is split into s sublots, and $x_{i,k}$ is the size of the i th sublot that completes its k th operation. Sublots of a job-lot can be immediately transferred from one machine to another for their next operation without waiting to complete other sublots. The goal is to determine the size and schedule of all sublots to minimize the makespan, which is the time to complete the third operation of the sublot processed as the last.

The assumptions made for the single-job problem are summarized here:

- The sublots of a job lot are processed without any interruption on every machine. i.e., pre-emption is not allowed.
- Each machine is ready at the beginning, say time zero, of the planning horizon. i.e., machines are not batching machines.
- At any time, only one item of a job lot can be processed by a machine.
- An unlimited storage space exists between the machines.
- An idle time on a machine may occur between processing sublots.

- Transfer times from one machine to another are negligibly so short and thus ignored.
- Setup times before processing the job-lot on a machine are negligibly so short and thus ignored.
- The number of sublots is known in advance and fixed from one machine to another.
- Processing times are known and deterministic.

3.1. Machine 2 is the primary machine

We first consider that M_2 is the primary machine where the third operation is performed. We investigate the problem for cases with consistent and variable sublots.

3.1.1. Consistent sublots

When the lot streaming is applied, sometimes there might be no advantage to change the size of the sublots after they have completed their processing on a machine. In this situation, it is reasonable to let the sublot sizes be constant (*consistent*) over all pairs of operations, i.e., $x_{i,k} = x_i$ for $i = 1,2,...,s$; $k = 1,2,3$ where $\sum_{1 \le i \le s} x_i = U$. ***Sublot availability*** assumption is used when sublots are consistent. i.e., a sublot can be processed at the next machine if all items in this sublot are completed on the current machine.

We start our analysis with the following lemma.

Lemma 1. *For the single-job problem where M_2 is the primary machine, it is sufficient to consider schedules of sublots where the last two operations of the sublots are processed consecutively on M_2.*

Proof. Consider any schedule of the consistent sublots. Suppose that there is a pair of sublots u and v, where the second operation of sublot v is immediately processed before the third operation of sublot u on M_2. Then interchanging the positions of sublots u and v on M_2 is feasible and does not increase makespan. On machine M_2, when all sublots, which are immediately processed before sublot u, are pair-wise interchanged with the second operation of sublot u, then it is possible to consecutively process the second and the third operations of sublot u on M_2. Similarly, it is possible to schedule consecutively the second and the third operations of all sublots on M_2.

From Lemma 1, the single-job problem can be illustrated by a network, as shown in Figure 1. Let $\bar{x} = (x_1, x_2, ..., x_s)$ denote the sublot sizes, and (k,i) be a node for the pair with operation k $(k = 1,2,3)$ and sublot i $(i = 1,...,s)$ where $p_k x_i$ is the processing time of the k th operation of sublot i. The vertical arc from node $(1, i)$ to node $(2, i)$ indicates that sublot i cannot be processed on M_2 unless it is completed on M_1. The horizontal arc from node $(1, i)$ to node $(1, i+1)$ indicates that M_1 can start to process sublot $i+1$ upon the completion of sublot i on M_1. Similarly, the horizontal arc from $(2, i)$ to $(3, i)$ represents that the third operation of sublot i can be started when its second operation is completed on M_2.

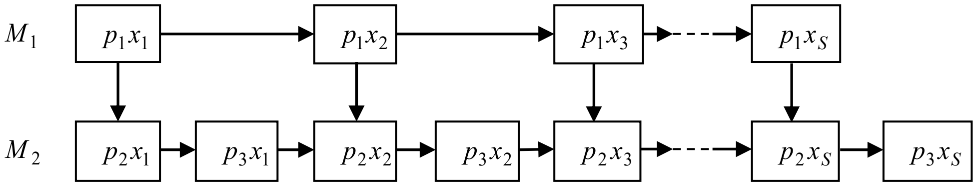

Figure 1. Network representation for the case where M_2 is the primary machine

Here, the goal is to determine the sublot sizes minimizing the ***critical path*** (longest path) length from node $(1, 1)$ to $(3, s)$, in which the sum of the processing times of the nodes on the critical path gives the length of the critical path.

The following theorem gives the optimal sublot sizes for the single-job problem where M_2 is the primary machine.

Theorem 1. *For the single-job problem where M_2 is the primary machine, the optimal consistent-sublot sizes are $x_1 = U/(1+\alpha+\alpha^2+\ldots+\alpha^{s-1})$, $x_i = \alpha^{i-1}x_1$ for $i = 2,\ldots,s$ where $\alpha = (p_2+p_3)/p_1$, $U = \sum_{1\le i\le s} x_i$, and the associated optimal makespan is $C_{\max} = p_1x_1 + (p_2+p_3)\sum_{i=1}^{s} x_i = \left(p_1/(1+\alpha+\alpha^2+\ldots+\alpha^{s-1}) + (p_2+p_3)\right)U$.*

Proof. See the Appendix A. ■

Theorem 1 proves that the single-job problem where M_2 is the primary machine is equivalent to the single-job problem in the basic two-machine flow shop with processing times p_1 and p_2+p_3 on M_1 and M_2, respectively. From Theorem 1, it is clear that the optimal consistent-sublot sizes can be determined in $O(s)$ time.

Example 1. In this numerical example, we illustrate Theorem 1, in which the consistent sublots are optimal. Suppose that we have a job lot of 70 identical items that will be split into three sublots, and the processing times for its three operations are 2, 3, and 1 time-units, respectively. Then, from Theorem 1, we determine that the sublot sizes are found to be 10, 20, and 40 units for the first, second, and third sublots, respectively. That is, $x_1 = U/(1+\alpha+\alpha^2+\ldots+\alpha^{s-1}) = 70/(1+2+2^{3-1}) = 10$, $x_2 = 2^1 x_1 = (2)(10) = 20$, and $x_3 = 2^2 x_1 = (4)(10) = 40$, where $\alpha = (p_2+p_3)/p_1 = (3+1)/2 = 2$. The optimal makespan is $C_{\max} = p_1x_1 + (p_2+p_3)\,U = (2)(10)+(3+1)(70) = 300$, as illustrated in Figure 2.

3.1.2. Variable sublots

In some cases, there might be some advantages to change the size of the sublots after they have completed their processing on a machine. Thus, variable sublots are preferred to the consistent sublots, using the ***item availability*** assumption where an individual item in a sublot can be processed at the next machine when this item is completed on the current machine.

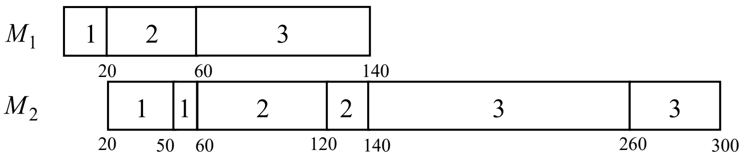

Figure 2. Optimal schedule of the sublots in Example 1

Remark 1. *There is no need to investigate the optimal solution for the single-job problem having variable sublots in the two-machine re-entrant flow shop where M_2 is the primary machine. The solution of the single-job problem having consistent sublots is also optimal for the single-job problem having variable sublots since only one set of sublot transfers from M_1 to M_2 is needed when the second and third operations are performed on M_2.*

3.2. Machine 1 is the primary machine

We now consider that M_1 is the primary machine where the third operation is performed. We again investigate the problem for cases with consistent and variable sublots.

3.2.1. Consistent sublots

Similar to the analysis given by Wang et al. (1997) for the basic two-machine re-entrant flow shop makespan minimization problem without lot streaming, we present the following lemma and theorem for finding the optimal solution to our problem having consistent sublots when M_1 is the primary machine. We first give the following definitions.

Definition 1. *A schedule is called a* ***compact schedule*** *if the first operations of all sublots are scheduled successively on M_1 and then followed by the third operations of all sublots.*

Definition 2. *A schedule is called a* ***permutation schedule*** *if all sublots are processed in the same order on both machines.*

Lemma 2. *For the single-job problem where M_1 is the primary machine, it is sufficient to consider only compact and permutation schedules of the sublots.*

Proof. Consider any feasible schedule σ with consistent sublots. We first show that σ can be transformed into a compact schedule on M_1 without worsening the makespan. Suppose we have a pair of two sublots, u and v, where the last (third) operation of sublot u immediately precedes the first operation of sublot v on M_1, i.e., $O_{u,3} \preceq O_{v,1}$. Then interchanging their positions does not worsen the makespan.

When all sublots, which immediately follow sublot u, are pair-wise interchanged with sublot u, we assume that all first operations of the sublots are scheduled first on M_1. If any, we may eliminate the idle time between the first operations of any pair of successive sublots by moving the start time of the second sublot in the pair to the left. This movement does not affect the feasibility and does not change the makespan. Now, assume that an idle time exists between the last operations of any pair of successive sublots. Then we can eliminate it by moving all sublots except the last to the right. Again, this movement does not affect the feasibility and does not change the makespan. This shows that the first and the last operations of the job-lot are performed continuously without idle time between the sublots on M_1. Now, consider any optimal schedule σ, which is compact on M_1 but in which the processing order of the sublots on the first pair (M_1, M_2) of machines is different. Let sublots u and v be the first pair of sublots such that $O_{u,1} \underline{\pi} O_{v,2}$ and $O_{v,2} \underline{p} O_{u,2}$. Let sublots u and v be the last pair of sublots such that $O_{u,1} \underline{\pi} O_{v,1}$ and $O_{v,2} \pi O_{u,2}$. Interchanging the order of $O_{u,2}$ and $O_{v,2}$ is possible and maintains compactness on M_1, and the makespan remains unchanged. This indicates that an optimal schedule, which is compact and the processing order of the sublots on the last pair (M_2, M_1) of machines is the same, exists.

Theorem 2. *Let $C_{\max}$ and $C'_{\max}$ denote the optimal makespan values of the single-job problem having consistent sublots in the two-machine re-entrant flow shop where M_1 is the primary machine and the single-job problem having consistent sublots in the basic three-machine flow shop, respectively. If $C'_{\max} > \sum_{i=1}^{s}(p_1 + p_3)x_i = (p_1 + p_3)U$ then $C_{\max} = C'_{\max}$; otherwise, $C_{\max} = (p_1 + p_3)U$.*

Proof. The single-job problem where M_1 is the primary machine can be represented by a network, as illustrated in Figure 3. Let $\bar{x} = (x_1, x_2, \ldots, x_s)$ denote the sublot sizes and (k,i) be a node for the pair with operation k $(k = 1,2,3)$ and sublot i $(i = 1,\ldots,s)$, having a weight of $p_k x_i$ that represents the sublot processing time. The vertical arc from node $(1,i)$ to node $(2,i)$ indicates that sublot i can be processed on M_2 upon its completion on M_1. The horizontal arc from node $(1,i)$ to node $(1,i+1)$ indicates that M_1 can start to process sublot $i+1$ upon the completion of sublot i on M_1. Similarly, the vertical arc from $(2,i)$ to $(3,i)$ represents that the third operation of a sublot can be started when the second operation is completed on M_2. In view of Lemma 2, we observe that the precedence constraint of arc between $(1,s)$ and $(3,1)$ becomes redundant if $C'_{\max} > (p_1 + p_3)U$. Thus, an idle time on M_1 exists between the first operation of the last sublot and the third operation of the first sublot. However, if the arc between $(1,s)$ and $(3,1)$ is removed from the network $N(\bar{x})$, the new network then represents the lot streaming problem in the three-machine flow shop. Therefore, it is clear that $C_{\max} = \max\{C'_{\max}, (p_1 + p_3)U\}$. It is obvious that $C_{\max} = C'_{\max}$ when $C'_{\max} > (p_1 + p_3)U$. Thus in this case, the lot streaming problems in the basic three-machine flow shop and the two-machine re-entrant flow shop are equivalent. However, $C_{\max} = (p_1 + p_3)U$ when $C'_{\max} \le (p_1 + p_3)U$. ■

As a result of Theorem 2, an optimal solution to the single-job problem where M_1 is the primary machine can be constructed by the optimal solution proposed by Glass et al. (1994) for the single-job problem having consistent sublots in the basic three-machine flow shop, as in the following corollary.

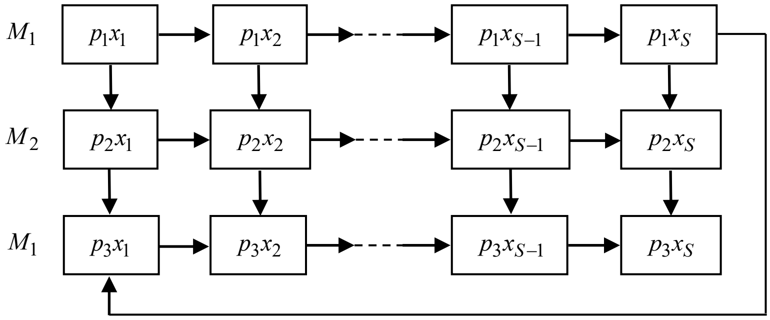

Figure 3. Network representation for the case where M_1 is the primary machine

Corollary 2. *When the makespan is minimized, the optimal consistent-sublot sizes for the single-job problem in the basic three-machine flow shop are also optimal for the single-job problem having consistent sublots in the two-machine re-entrant flow shop where M_1 is the primary machine, and they are as follows:*

(i) *If $p_2^2 \le p_1 p_3$, then the optimal sublot sizes are:*

$$x_1 = \begin{cases} U(\beta - 1)/(\beta^s - 1) & \text{if } p_1 \ne p_3 \\ U/s & \text{if } p_1 = p_3 \end{cases}, \quad x_i = \beta^{i-1} x_1 \text{ for } 2 \le i \le s \text{ where}$$

$\beta = (p_2 + p_3)/(p_1 + p_2)$.

(ii) *If $p_2^2 > p_1 p_3$, then there exists a crossover sublot h, which can be determined by a search algorithm in $O(\log s)$ time, for which the optimal sublot sizes are:*

$$x_h = \begin{cases} U/\{(\alpha^h - 1)/(\alpha - 1) + (\gamma^{s-h+1} - 1)/(\gamma - 1) - 1\} & \text{if } p_1 \ne p_2, p_2 \ne p_3 \\ U/\{h - 1 + (\gamma^{s-h+1} - 1)/(\gamma - 1)\} & \text{if } p_1 = p_2, p_2 \ne p_3 \\ U/\{(\alpha^h - 1)/(\alpha - 1) + s - h\} & \text{if } p_1 \ne p_2, p_2 = p_3 \end{cases},$$

$x_i = \alpha^{h-i} x_h$ *for* $1 \le i \le h-1$, $x_i = \gamma^{i-h} x_h$ *for* $h \le i \le s$ *where* $\alpha = p_1 / p_2$ *and* $\gamma = p_3 / p_2$.

Example 2. Assume that we have a job lot of 70 identical items that will be split into three sublots, and the processing times for its three operations are 1, 4, and 2 time-units, respectively. This case corresponds to the second case in Corollary 2 since $p_2^2 = (4)^2 = 16 > p_1 p_3 = (1)(2) = 2$. The size of the crossover sublot h is determined by $x_h = U/\{(\alpha^h - 1)/(\alpha - 1) + (\gamma^{s-h+1} - 1)/(\gamma - 1) - 1\}$ where $\alpha = p_1/p_2 = 1/4$ and $\gamma = p_3/p_2 = 1/2$ since $p_1 = 1 \neq p_2 = 4 \neq p_3 = 2$.

When $h = 1$, $x_1 = U/\left(\frac{\alpha^h - 1}{\alpha - 1} + \frac{\gamma^{s-h+1} - 1}{\gamma - 1} - 1\right) = 70/\left(\frac{(1/4)^1 - 1}{(1/4) - 1} + \frac{(1/2)^{3-1+1} - 1}{(1/2) - 1} - 1\right) =$ $\frac{70}{1.75} = 40$, $x_2 = \gamma^{2-1}x_1 = (1/2)^1(40) = 20$, $x_3 = \gamma^{3-1}x_1 = (1/2)^2(40) = 10$, and $C'_{\max} = 340$.

When $h = 2$, $x_2 = U/\left(\frac{\alpha^h - 1}{\alpha - 1} + \frac{\gamma^{s-h+1} - 1}{\gamma - 1} - 1\right) = 70/\left(\frac{(1/4)^2 - 1}{(1/4) - 1} + \frac{(1/2)^{3-2+1} - 1}{(1/2) - 1} - 1\right) =$ $\frac{70}{1.75} = 40$, $x_1 = \alpha^{2-1}x_1 = (1/4)^1(40) = 10$, $x_3 = \gamma^{3-2}x_1 = (1/2)^1(40) = 20$, and $C'_{\max} = 330$ (See Figure 4a).

When $h = 3$, $x_3 = U/\left(\frac{\alpha^h - 1}{\alpha - 1} + \frac{\gamma^{s-h+1} - 1}{\gamma - 1} - 1\right) = 70/\left(\frac{(1/4)^3 - 1}{(1/4) - 1} + \frac{(1/2)^{3-3+1} - 1}{(1/2) - 1} - 1\right) \cong$ 53.33, $x_1 = \alpha^{3-1}x_1 = (1/4)^2(53.33) = 3.33$, $x_2 = \alpha^{3-2}x_1 = (1/4)^1(53.33) = 13.33$, and $C'_{\max} = 390$.

It is clear that the optimal sublots are achieved when $h = 2$, and their sizes are $x_1 = 10$, $x_1 = 40$, and $x_3 = 20$. Note that the optimal makespan is $C_{\max} = C'_{\max} = 330$ since $C'_{\max} = 330 > (p_1 + p_3)U = (1+2)(70) = 210$, and the optimal schedule is obtained by carrying the third operations of the sublots in the three-machine flow shop problem to M_1, as illustrated in Figure 4b.

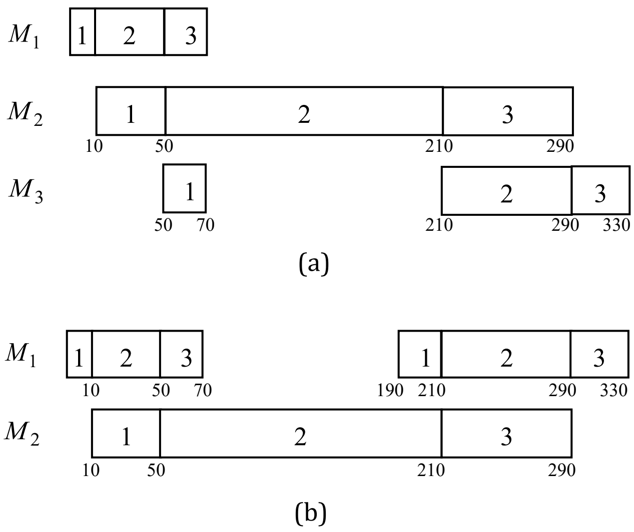

Figure 4. Optimal schedules with consistent sublots for the single-job problem in Example 2: (a) Three-machine flow shop; (b) Two-machine re-entrant flow shop.

3.2.2. Variable sublots

As a result of Theorem 2, an optimal solution to the single-job problem having variable sublots where M_1 is the primary machine can be constructed by the optimal solution proposed by Trietsch & Baker (1993) for the single-job problem with variable sublots in the basic three-machine flow shop, as in the following corollary.

Corollary 3. *When the makespan is minimized, the optimal variable-sublot sizes in the single-job problem in the basic three-machine flow shop are also optimal for the single-job problem having the variable sublots in the two-machine re-entrant flow shop where M_1 is the primary machine, and they are as follows:*

(i) *If $p_2^2 \le p_1 p_3$, then the consistent sublots are optimal, and they are:* $x_1 = U(\beta - 1)/(\beta^s - 1)$, $x_i = \beta^{i-1} x_1$ *for* $2 \le i \le s$ *where* $\beta = (p_2 + p_3)/(p_1 + p_2)$.

(ii) *If $p_2^2 > p_1 p_3$, then the variable sublots are optimal,*

- *the optimal sublot sizes between M_1* and *M_2 are:* $x_1 = U(\alpha - 1)/(\alpha^s - 1)$, $x_i = \alpha^{i-1} x_1$ *for* $2 \le i \le s$ *where* $\alpha = p_2 / p_1$, *and*
- *the optimal sublot sizes between M_2* and *M_1 are:* $x_1 = U(\gamma - 1)/(\gamma^s - 1)$, $x_i = \gamma^{i-1} x_1$ *for* $2 \le i \le s$ *where* $\gamma = p_3 / p_2$.

From Corollary 3, it is clear that the optimal variable-sublot sizes can be determined in $O(s)$ time.

Example 3. In this numerical example, we illustrate the second case of Corollary 3, in which the variable sublots are optimal since $p_2^2 > p_1 p_3$. Assume that we have a job lot of 15 identical items that will be split into two sublots, and the processing times for its three operations are 1, 2, and 1 time-units, respectively. From Corollary 2, the sublot sizes between M_1 and M_2 are found to be $x_1 = 15(2-1)/(2^2 - 1) = 5$ and $x_2 = (2^1)(5) = 10$ where $\alpha = p_2 / p_1 = 2/1 = 2$ and $p_2^2 = (2)^2 = 4 > p_1 p_3 = (1)(1)$. (See Figure 5a). Similarly, the sublot sizes between M_2 and M_1 are calculated as $x_1 = 15((1/2) - 1)/((1/2)^2 - 1) = 10$ and $x_2 = ((1/2)^1)(10) = 5$ since $\gamma = p_3 / p_2 = 1/2$. Thus, the optimal makespan of the problem having variable sublots in the basic three-machine flow shop is 40, and the optimal schedule of the problem in the two-machine re-entrant flow shop is obtained by carrying the third operations of the sublots in the optimal schedule of the problem in the three-machine flow shop to M_1, as illustrated in Figure 5b.

4. Multi-job case

In this section, we extend our problem presented to the multi-job case. The multi-job problem is explained as follows: There is a job-lot set $J = \{j \mid j = 1,2,\ldots,n\}$ where each job-lot has U_j identical items of type j. Let $p_{j,k}$ be the processing time for

k th $(k=1,2,3)$ operation of job-lot j. There are two machines M_1 and M_2. Each job-lot is processed first on M_1, on M_2 for its second operation before returning to the primary machine (M_1 or M_2) for the third operation. Suppose that job-lot j is split into s_j sublots, then our goal is to determine the sublot sizes for each job-lot and the schedule of all sublots and job lots to minimize the makespan.

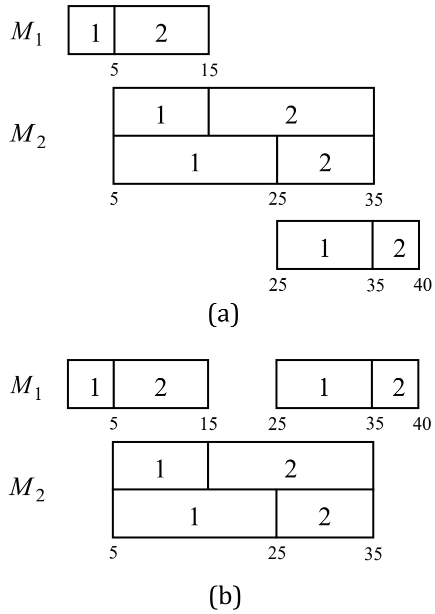

Figure 5. Optimal schedules with variable sublots for the single-job problem in Example 3: (a) Three-machine flow shop; (b) Two-machine re-entrant flow shop.

For the multi-job problem, we consider all assumptions made for the single-job problem. Furthermore, we do not allow the ***intermingling*** of different job lots. That is, once a sublot of a job-lot is started on a machine, all other job lots should wait until all of the remaining sublots of that job-lot are completed on the same machine. Note that intermingling the sublots belonging to different job lots may further reduce the makespan, but we focus on the no-intermingling case in our study. Intermingling can be considered in a future study as future research, as we point out in Section 5.

As in the case of the single-job problem, we investigate two cases associated with the primary machine.

4.1. Machine 2 is the primary machine

The following lemma gives the basic structure of the job-sequence for the multi-job problem where M_2 is the primary machine.

Lemma 3. *For an optimal solution of the multi-job problem where M_2 is the primary machine, it is sufficient to consider the job-sequence in which the last two operations of each sublot belonging to a job-lot are processed successively on M_2.*

Proof. Omitted since it is similar to the proof of Lemma 1. ■

The multi-job problem where M_2 is the primary machine can be decomposed into two sub-problems: (a) the ***job-sequencing*** and (b) the ***sublot-sizing***.

4.1.1. Job-sequencing sub-problem

The multi-job problem reduces to the determination of the optimal sequence of job lots when the sublot-sizing sub-problem for each job-lot has already been solved. That is, the job-sequencing sub-problem needs to be solved. Suppose that each job-lot j is independently split into sublots by Corollary 2 or Corollary 3. Let

RI_j = time lag between the start of the first operation on M_1 and the latest start time of the second operation on M_2 for job-lot j. i.e., the latest delay time, simply called run-in-delay for job-lot j on M_2 without affecting the completion time of job-lot j (called as ***run-in delay for Operation 2***).

RO_j = time lag between the completion times of the first and the second operations for job-lot j (called as ***run-out delay for Operation 2***).

RI'_j = time lag between the latest start of the second operation on M_2 and the latest start time of the third operation on M_1 for job-lot j. i.e., the latest delay time, simply called run-in-delay for job-lot j on M_1 without affecting the completion time of job-lot j (called as ***run-in delay for Operation 3***).

RO'_j = time lag between the completion times of the second and the third operations for job-lot j (called as ***run-out delay for Operation 3***).

The following theorem provides the optimal solution to the job-sequencing sub-problem.

Theorem 3. *Given the solution of the sublot-sizing sub-problem, job-lot v precedes job-lot z in an optimal job-sequence if $\min\{RI_v, RO_z\} \le \min\{RI_z, RO_v\}$, where RI_j = $\min_{1\le i\le s_j}\left\{\sum_{r=1}^{i} p_{j,1}x_{j,r} - \sum_{r=1}^{i-1}(p_{j,2}+p_{j,3})x_{j,r}\right\}$, and $RO_j = RI_j + (p_{j,2}+p_{j,3}-p_{j,1})U_j$ for any job-lot j.*

Proof. See the Appendix B. ■

4.1.2. Sublot-sizing sub-problem

The sublot-sizing sub-problem needs to be solved for each job-lot when the job-sequencing sub-problem has already been solved. The following theorem provides the optimal solution to the sublot-sizing sub-problem.

Theorem 4. *Given the solution of the job-sequencing sub-problem, the optimal sublot sizes in a job-lot processed in position* $[j]$ *of the job-sequence are* $x_{[j],1} = U_{[j]}/(1+\alpha+\alpha^2 +\ldots+\alpha^{s[j]-1})$, $x_{[j],i} = \alpha^{i-1}x_{[j],1}$ *for* $i = 2,\ldots,s_{[j]}$ *where* $\alpha = (p_{[j],2} + p_{[j],3})/p_{[j],1}$.

Proof. See the Appendix C. ■

From Theorems 3 and 4, we have the following result.

Corollary 4. *Solving the sublot-sizing sub-problem using Theorem 4 and then solving the job-sequencing sub-problem using Theorem 3 provides the optimal solution to the multi-job problem where* M_2 *is the primary machine.*

Based on Corollary 4, we propose the following exact-solution algorithm with a computational complexity of $O(n \log n)$ to solve the multi-job problem where M_2 is the primary machine and n is the total number of job lots.

Algorithm 1.

Step 1: [*Sublot-sizing*] Calculate the size for each sublot of the job-lot in the set $J = \{ j \mid j = 1, 2, \ldots, n\}$ as $x_{j,1} = U_j / (1+\alpha+\alpha^2+\ldots+\alpha^{s_j-1})$ and $x_{j,i} = \alpha^{i-1}x_{j,1}$ for $i = 2,\ldots,s_j$ where $\alpha = (p_{j,2} + p_{j,3})/p_{j,1}$.

Step 2: [*Job-sequencing*]

(a) Set $RI_j = p_{j,1}U_j/(1+\alpha+\alpha^2+\ldots+\alpha^{s_j-1})$ and $RO_j = RI_j + (p_{j,2} + p_{j,3} - p_{j,1})U_j$.

(b) To obtain the job-sequence $\pi = \{\pi[j] \mid j = 1, 2, \ldots, n\}$, consider all jobs in the job-lot set J and apply Johnson's Algorithm with processing times on M_1 and M_2 replaced by RI_j and RO_j, respectively.

(ii) Calculate the associated makespan as

$$C_{\max} = \max_{1\le w\le n}\left(\sum_{j=1}^{w} RI_{\pi[j]} - \sum_{j=1}^{w-1} RO_{\pi[j]}\right) + \sum_{j=1}^{n}(p_{\pi[j],2} + p_{\pi[j],3})U_{\pi[j]}$$

4.2. Machine 1 is the primary machine

This section considers the multi-job problem where M_1 is the primary machine. Unfortunately, this problem is more complicated than the multi-job problem where M_2 is the primary machine discussed in Section 4.1.

Theorem 5. *The multi-job problem where* M_1 *is the primary machine is NP-hard in the strong sense.*

Proof. Suppose that each job-lot has one sublot only. This special case of our multi-job problem is equivalent to the multi-job problem without lot streaming in the basic two-machine re-entrant flow shop where M_1 is the primary machine. It has been proven by Wang et al. (1997) that this special case is *NP*-hard in the strong sense. Thus, our multi-job problem where M_1 is the primary machine is also strongly *NP*-hard. ■

The following lemma restricts our search to the compact and permutation schedules of the job lots for the optimal schedule of the multi-job problem where M_1 is the primary machine.

Lemma 4. *For an optimal solution of the multi-job problem where M_1 is the primary machine, it is sufficient to consider only compact and permutation schedules of the job lots.*

Proof. Omitted since it is similar to the proof of Lemma 2. ■

4.2.1. A Polynomial-time solvable case

Since the multi-job problem where M_1 is the primary machine has been shown to be *NP*-hard in Theorem 5, an optimal algorithm solving the problem in polynomial-time is impossible. Therefore, we first examine a polynomial-time solvable case of the problem that corresponds to the case, in which the sublots of all job lots are independently determined by Corollaries 2 or 3 and the idle time between the first and the third operations of all jobs on M_1 is zero.

Let I_j be the idle time on M_1 between the first and the third operations of job-lot j when it is independently split into sublots by Corollaries 2 or 3. The following theorem provides the optimal schedule for this special case.

Theorem 6. *If $I_j = 0$ for every job-lot j, then arbitrarily sequencing all jobs as illustrated in Figure 6 is an optimal schedule for the multi-job problem where M_1 is the primary machine.*

Proof. If $I_j = 0$ for every job-lot j, there will be no idle time between the first and the third operations of job-lot j on M_1 when job lots are arbitrarily sequenced. Then, the time to complete all operations of job-lot j becomes $T_j = (p_{j,1} + p_{j,3})U_j + I_j = (p_{j,1} + p_{j,3})U_j$ when it is independently split into sublots, and $C_{\max} = \sum_{j=1}^{n} T_j = \sum_{j=1}^{n} (p_{j,1} + p_{j,3})U_j$ becomes the optimal makespan.

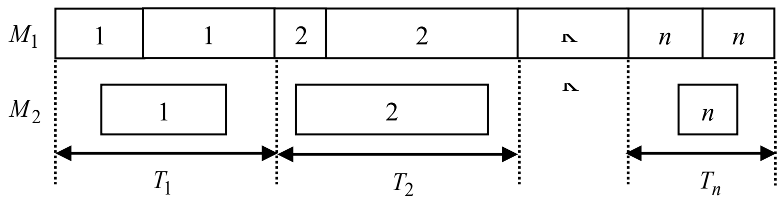

Figure 6. Optimal schedule obtained by Theorem 6

4.2.2. Proposed heuristic algorithm

For a given job-sequence π, we define the job-lot $\pi[d]$ such that $C_{\pi[d-1],2} \le \sum_{j=1}^{n} p_{j,1}U_j < C_{\pi[d],2}$ where $C_{\pi[d],2}$ is the time to complete the second operation of job-lot $\pi[d]$, and $C_{\pi[d],2} = C_2(\pi) - \sum_{j=d+1}^{n} p_{\pi[j],2}U_{\pi[d]} = \max_{j=1,..,n}\left(\sum_{i=1}^{j} RI_{\pi[i]} - \sum_{i=1}^{j-1} RO_{\pi[i]}\right) + \sum_{j=1}^{n} p_{\pi[j],2}U_{\pi(j)}$ where $C_2(\pi)$ is the completion time of all job lots in sequence π on M_2, and $C_{\pi[d-1],2} = C_{\pi[d],2} - p_{\pi[d],2}U_{\pi[d]}$. Here job-lot $\pi[d]$ that will be used to develop the proposed algorithm is called a ***partition job-lot*** and is the first job whose second operation finishes later than the completion of all first operations on M_1 (see Figure 7). From the definition of the job-lot $\pi[d]$, it is clear that no idle time exists between job lots following job-lot $\pi[d]$ on M_2. Thus, permutation sequence π is partitioned into three subsequences: $J_B = \{\pi[j] \mid j = 1,..., \ d-1\}$, $J_D = \{\pi[j] \mid j = d\}$, and $J_A = \{\pi[j] \mid j = d+1,,...,n\}$.

We propose the following heuristic algorithm with a computational complexity of $O(n \log n)$ to solve the multi-job problem where M_1 is the primary machine.

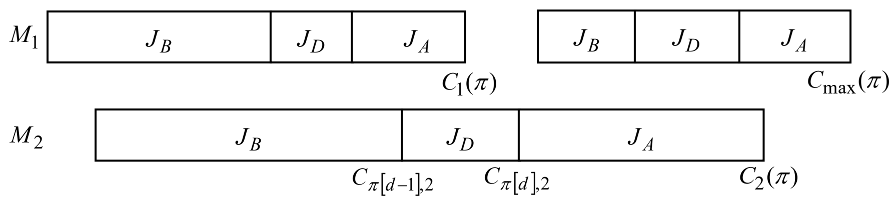

Figure 7. Partition job-lot in position d of the job-sequence π

Algorithm 2.

Step 1: For each job lot j,

(a) If sublots are considered as consistent, then use the sublot sizes in Corollary 2; otherwise (variable sublots), use the sublot sizes in Corollary 3.

(b) Compute the makespan T_j, and idle time on M_1 as $I_j = T_j - (p_{j,1} + p_{j,3})U_j$.

Step 2: Divide the job-lot set J into two sets: $J^1 = \{j \mid I_j = 0\}$ and $J^2 = \{j \mid I_j > 0\}$.

Step 3: If $J^2 = \varnothing$, then any arbitrary sequence of job lots is optimal, and the optimal makespan is $C^*_{\max} = \sum_{j=1}^{n}(p_{j,1} + p_{j,3})U_j$, stop; otherwise, go to Step 4.

Step 4: Compute the run-in delays (RI_j and RI'_j) and run-out delays (RO_j and RO'_j). Schedule all job lots $J = \{j \mid j = 1,\dots,n\}$ by applying Johnson's Algorithm with processing times on M_1 and M_2 replaced by RI_j and RO_j to obtain the job-sequence $\pi = \{\pi[j] \mid j = 1,\dots,n\}$.

Step 5: In the job-sequence π, determine the partition job-lot $\pi[d]$ such that $C_{\pi[d-1],2} \le \sum_{j=1}^{n} p_{j,1}\, U_j < C_{\pi[d],2}$ where $C_{\pi[d],2} = \max_{j=1,\dots,n}\left(\sum_{i=1}^{j} RI_{\pi[i]} - \sum_{i=1}^{j-1} RO_{\pi[i]}\right) + \sum_{j=1}^{d} p_{\pi[j],2} U_{\pi(j)}$, and $C_{\pi[d-1],2} = C_{\pi[d],2} - p_{\pi[d],2}\, U_{\pi[d]}$.

Step 6: Compute the associated makespan $C_{\max}(\pi)$ as:

$$C_{\max}(\pi) = \max\Big\{\sum_{j=1}^{n} p_{\pi[j],1} U_{\pi[j]} + \sum_{j=1}^{d-1} p_{\pi[j],3} U_{\pi[j]},\ \max_{j=1,\dots,n}\Big\{\sum_{i=1}^{j} RI_{\pi[i]} - \sum_{i=1}^{j-1} RO_{\pi[i]}\Big\} + \sum_{j=1}^{d-1} p_{\pi[j],2} U_{\pi[j]} + \max_{j=d,\dots,n}\Big\{\sum_{i=d}^{j} RI'_{\pi[i]} - \sum_{i=d}^{j-1} RO'_{\pi[i]}\Big\}\Big\} + \sum_{j=d}^{n} p_{\pi[j],3} U_{\pi[j]}$$

Step 7: If $C_{\max}(\pi) = \sum_{j=1}^{n} (p_{j,1} + p_{j,3})\, U_j$, then the job-sequence π is optimal, stop; otherwise, go to Step 8.

Step 8: Consider the job lots in $J_D \cup J_A = \{\pi[j] \mid j = d,\dots,n\}$, and apply Johnson's Algorithm with processing times on M_1 and M_2 replaced by RI'_j and RO'_j to obtain the partial job-sequence $\sigma = \{\sigma[j] \mid j = 1,\dots,n-d+1\}$.

Step 9: Set the final sequence of job lots as $\varphi = \{\pi, \sigma\}$ where $\pi = \{\pi[j] \mid j = 1,\dots,d-1\}$ followed by $\sigma = \{\sigma[j] \mid j = 1,\dots,n-d+1\}$. The associated makespan is

$$C_{\max}(\phi) = \max\Big\{\sum_{j=1}^{n} p_{\pi[j],1} U_{\pi[j]} + \sum_{j=1}^{d-1} p_{\pi[j],3} U_{\pi[j]},\ \max_{j=1,\dots,n}\Big\{\sum_{i=1}^{j} RI_{\pi[i]} - \sum_{i=1}^{j-1} RO_{\pi[i]}\Big\} + \sum_{j=1}^{d-1} p_{\pi[j],2} U_{\pi[j]} + \max_{j=d,\dots,n}\Big\{\sum_{i=d}^{j} RI'_{\sigma[i]} - \sum_{i=d}^{j-1} RO'_{\sigma[i]}\Big\} + \sum_{j=d}^{n} p_{\sigma[j],3} U_{\sigma[j]}$$

4.2.2. Lower bounds on the makespan

We now develop four lower bounds on the makespan, and then we take the maximum of these lower bounds as a global lower bound, which will be used to test Algorithm 2.

Lower bound 1. This lower bound is obtained by assuming that there will be no idle time between the first and the third operations of each job lot on M_1. Thus, the natural lower bound is

$$LB_1 = \sum_{j=1}^{n} (p_{j,1} + p_{j,3}) U_j \ , \tag{1}$$

which is equivalent to the total processing time of all job lots on M_1.

Lower bound 2. We derive the second lower bound as

$$LB_2 = \min_{j=1,..,n} \left(RI_j \right) + \sum_{j=1}^{n} p_{j,2} U_j + \min_{j=1,..,n} \left(RO'_j \right) \tag{2}$$

by assuming that the job lot with the minimum run-in delay for Operation 2 is processed as the first job-lot in the job-sequence, M_2 operates continuously without any idle time between job lots, and the job lot with the minimum run-out delay for Operation 3 is processed as the last job lot in the job-sequence.

Lower bound 3. To derive our third lower bound, we assume that all job lots do not wait for their operations to be performed on M_2. For a sequence π, we can establish the third lower bound as

$$LB_3 = \max_{j=1,..,n} \left(\sum_{i=1}^{j} RI_{\pi[i]} + \sum_{i=1}^{j-1} \Delta_{\pi[i]}^{1,2} + \sum_{i=j}^{n} p_{\pi[i],2} \, U_{\pi(i)} \right) + \min_{j=1,..,n} \left(RO'_j \right) \tag{3}$$

where $\pi[i]$ is the job-lot in position i of the job-sequence π, and $\Delta_{\pi[i]}^{1,2}$ is the overlapping time for job lot $\pi[i]$ when both M_1 and M_2 are busy (i.e., operating continuously without idle time between sublots), as illustrated in Figure 8.

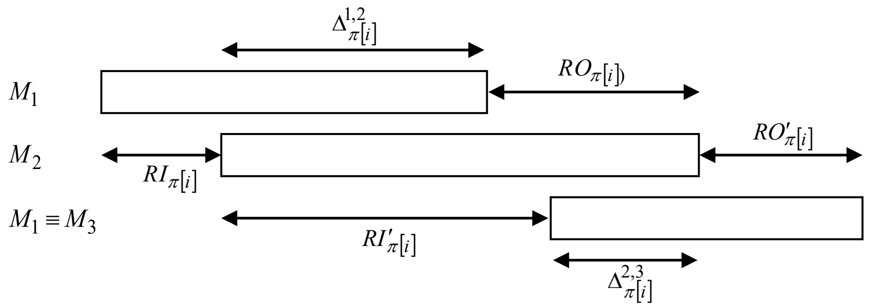

Figure 8. Run-in and run-out delays

We can rewrite (3) as

$$\begin{aligned} LB_3 &= \max_{j=1,..,n} \left\{ \sum_{i=1}^{j} RI_{\pi[i]} + \sum_{i=1}^{j-1} (p_{\pi[i],2} U_{\pi[i]} - RO_{\pi[i]}) + \sum_{i=j}^{n} p_{\pi[i],2} U_{\pi[i]} \right\} + \min_{j=1,..,n} \left\{ RO'_j \right\} \\ &= \max_{j=1,..,n} \left\{ \sum_{i=1}^{j} RI_{\pi[i]} + \sum_{i=1}^{j-1} p_{\pi[i],2} U_{\pi[i]} - \sum_{i=1}^{j-1} RO_{\pi[i]} + \sum_{i=j}^{n} p_{\pi[i],2} U_{\pi[i]} \right\} + \min_{j=1,..,n} \left\{ RO'_j \right\} \\ &= \max_{j=1,..,n} \left\{ \sum_{i=1}^{j} RI_{\pi[i]} - \sum_{i=1}^{j-1} RO_{\pi[i]} \right\} + \sum_{j=1}^{n} p_{[i],2} U_{[i]} + \min_{j=1,..,n} \left\{ RO'_j \right\} \end{aligned} \tag{4}$$

Note that the last two terms in (4) are constant and independent from the sequence, and the first term, $\max_{j=1,..,n} \left(\sum_{i=1}^{j} RI_{\pi[i]} - \sum_{i=1}^{j-1} RO_{\pi[i]} \right)$, gives the total idle time on M_2 before completing the second operations of all job lots. The first term equals Johnson's expression in which the processing times on M_1 and M_2 are replaced by the run-in and run-out delays, respectively. Thus, the job-sequence π is obtained by job-lot k preceding job-lot l if $\min\{RI_k, RO_l\} \le \min\{RO_k, RI_l\}$.

Therefore, we obtain the third lower bound as

$$LB_3 = C^*_{\max}\left(JA(RI, RO)\right) + \min_{j=1,..,n}\left\{RO'_j\right\} \tag{5}$$

where $C^*_{\max}\left(JA(RI, RO)\right)$ is the makespan obtained by Johnson's Algorithm (*JA*) with processing times on M_1 and M_2 replaced by RI_j and RO_j, respectively.

Lower bound 4. For any job-sequence π, we can establish the fourth lower bound as

$$LB_4 = \min_{j=1,..,n}\left\{RI_j\right\} + \max_{j=1,..,n}\left\{\sum_{i=1}^{j} RI'_{\pi[i]} + \sum_{i=1}^{j-1} \Delta^{2,3}_{\pi[i]} + \sum_{i=j}^{n} p_{\pi[i],3} U_{\pi[i]}\right\} \tag{6}$$

where $\Delta^{2,3}_{\pi[i]}$ is the overlapping time of the second and third operations for job lot $\pi(i)$ as shown in Figure 8.

Equation (6) can be rewritten as

$$LB_4 = \min_{j=1,..,n}\left\{RI_j\right\} + \max_{j=1,..,n}\left\{\sum_{i=1}^{j} RI'_{\pi[i]} + \sum_{i=1}^{j-1} (p_{\pi[i],3} U_{\pi[i]} - RO'_{\pi[i]}) + \sum_{i=j}^{n} p_{\pi[i],3} U_{\pi[i]}\right\}$$

$$= \min_{j=1,..,n}\left\{RI_j\right\} + \max_{j=1,..,n}\left\{\sum_{i=1}^{j} RI'_{\pi[i]} - \sum_{i=1}^{j-1} RO'_{\pi[i]}\right\} + \sum_{i=1}^{n} p_{\pi[i],3} U_{\pi[i]} \tag{7}$$

Note that the second term in (7), $\max_{j=1,..,n}\left\{\sum_{i=1}^{j} RI'_{\pi[i]} - \sum_{i=1}^{j-1} RO'_{\pi[i]}\right\}$, is minimized by job-lot k preceding job lot l in the job-sequence if $\min\{RI'_k, RO'_l\} \le \min\{RO'_k, RI'_l\}$. Therefore, we obtain the fourth lower bound as

$$LB_4 = \min_{j=1,..,n}\left\{RI_j\right\} + C^*_{\max}\left(JA(RI', RO')\right) \tag{8}$$

where $C^*_{\max}\left(JA(RI', RO')\right)$ is the makespan obtained by Johnson's Algorithm with processing times on M_1 and M_2 replaced by RI'_j and RO'_j, respectively.

The global lower bound GLB becomes the maximum of the lower bounds developed above. i.e., $GLB = \max_{i=1,...,4}\{LB_i\}$.

Example 4. As an illustration of the proposed algorithm and the global lower bound, we consider the five-job problem in Table 1.

Table 1. Processing times, number of sublots, and lot sizes

j	$p_{j,1}$	$p_{j,2}$	$p_{j,3}$	s_j	U_j
1	3	2	3	4	40
2	1	2	2	3	30
3	1	2	7	2	20
4	1	4	2	3	70
5	2	2	1	3	35

When all job lots are independently split into their number of consistent sublots, the sublot sizes and T_j and I_j values are obtained, as illustrated in Table 2.

Table 2. Sublot sizes with T_j and I_j values

j	$x_{j,1}$	$x_{j,2}$	$x_{j,3}$	$x_{j,4}$	T_j	I_j
1	10	10	10	10	240	0
2	6	12	12	-	90	0
3	5	15	-	-	160	0
4	10	40	20	-	330	120
5	14	14	7	-	105	0

Job-lot set J is decomposed into two job-lot sets, $J^1 = \{1, 2, 3, 5\}$ and $J^2 = \{4\}$. We continue with Step 4 of Algorithm 2 since the job-lot set J^2 is not empty. The run-in and run-out delays for each job lot are obtained as shown in Table 3.

Table 3. Run-in and run-out delays

j	RI_j	RO_j	RI'_j	RO'_j
1	30	20	20	30
2	6	24	12	24
3	5	105	10	105
4	10	220	180	40
5	28	28	42	7

The application of Johnson's Algorithm with processing times on M_1 and M_2 replaced by RI_j and RO_j; respectively, gives the job-sequence $\pi = \{3, 2, 4, 5, 1\}$. The partition job-lot is job 2, and it is in the second position (i.e., $d = 2$) of the sequence π. The associated makespan $C_{\max}(\pi)$ for the sequence π is computed as 805. The global lower bound becomes $GLB = \max\{LB_1, LB_2, LB_3, LB_4\} = 805$ where

$$LB_1 = \sum_{j=1}^{n}(p_{j,1} + p_{j,3})U_j = (3+3)(40) + (1+7)(20) + (1+2)(70) + (2+1)(35) = 805,$$

$$LB_2 = \min_{j=1,..,n}\left(RI_j\right) + \sum_{j=1}^{n} p_{j,2}U_j + \min_{j=1,..,n}\left(RO'_j\right) = \min\{30, 6, 5, 10, 28\} + ((2)(40) + (2)(30)$$

$$+(4)(70) + (2)(35) + \min\{30, 24, 105, 40, 7\} = 542,$$

$$LB_3 = C^*_{\max}\left(JA\left(RI, RO\right)\right) + \min_{j=1,..,n}\left(RO'_j\right) = C^*_{\max}\left(4-2-3-1-5\right) + \min\{30, 24, 105, 40, 7\}$$

$= 535 + 7 = 542$,

$LB_4 = \min_{j=1,\ldots,n} \{RI_j\} + C^*_{\max}\left(JA\left(RI', RO'\right)\right) = \min\{30, 6, 5, 10, 28\} + C^*_{\max}\left(3-1-4-2-5\right)$

$= 5 + 560 = 565$.

The job-sequence $\pi = \{3, 2, 4, 5, 1\}$ is optimal since the associated makespan equals the natural lower bound LB_1. Thus, we stop.

4.2.3. Computational experiments and results

The efficiency measure of Algorithm 2 is the computational time required to solve the problem. However, its computational time is not measured provided since it is relatively very short, less than a few seconds. On the other hand, to test the effectiveness of Algorithm 2, we generate the parameters for our problem instances as follows:

n: number of job lots, $n \in \{5, 10, 15, 20, 25, 50, 75, 100\}$.

$p_{j,k}$: k th operation's processing time for job lot j is randomly generated form from a discrete uniform distribution over [1, 10], including the lower and upper limits.

s_j: number of sublots for any job lot j is randomly generated form from a discrete uniform distribution over [2, 10].

U_j: lot size for any job lot j is randomly generated form from a discrete uniform distribution over [2, 50].

For each possible number of job lots from 5 to 100, we first generate 100 problem instances in which the. The processing times for all operations are randomly distributed without any dominance of a specific operation, and a total of 800 problem instances are tested. The following statistics are collected:

NT_Z = number of times percent deviation is zero (i.e., the heuristic makespan equals the global lower bound).

$NT_{(0,1]}$ = number of times the percent deviation is greater than zero but less than or equal to 1 (i.e., $C^H_{\max} \le 1.01\,GLB$).

AVE = average percent deviation.

MAX = maximum percent deviation.

From the results of our experiments, we observed that Algorithm 2 finds the optimum makespan (i.e., the heuristic makespan equals the global lower bound) for all 800 test problems when all processing times are randomly generated without any dominance of a specific operation. That is, $NT_Z = 800$.

However, to evaluate the effectiveness of Algorithm 2 when the same operation for all job lots is dominant, we repeated our computational experiments with three data sets. The first data set, D1, assumes that the maximum processing time is on the first operation for all job lots, i.e., $p_{j,1} \ge \max\{p_{j,2}, p_{j,3}\}$ for $\forall j$. Similarly, the second and third data sets D2 and D3 correspond to the cases where $p_{j,2} \geq \max\{p_{j,1}, p_{j,3}\}$ and $p_{j,3} \geq \max\{p_{j,1}, p_{j,2}\}$ for $\forall j$, respectively.

In each of the data sets, we assume that the processing time of the dominant operation for each job-lot is randomly generated form from a discrete uniform distribution over [6, 10], and the processing times of the other operations are randomly generated form from a discrete uniform distribution over [1, 5]. Our experiments with data sets D2 and D3 show that the global lower bound equals the heuristic makespan in all 800 problem instances tested when either the first or third operation is dominant. This result means that the global lower bound is effective when the first or third operation is dominant. However, the same argument is not valid for the case, where the second operation is dominant, since the global lower bound equals the heuristic makespan in 336 problem instances out of 800, as illustrated in Table 4.

From Table 4, it is clear that the heuristic makespan deviates 1 percent from the global lower bound in 790 problem instances out of 800 for the case where the second operation is dominant. In the remaining ten problem instances when $n = 5$, the average and maximum deviations from the global lower bound are 0.352 percent and 2.055 percent, respectively. It should also be noticed that the average and maximum deviations decrease as the number of jobs increases.

Table 4. Performance of Algorithm 2 when the second operation is dominant

n	$NPIS$	NT_Z	$NT_{(0,1]}$	AVE	MAX
5	100	45	45	0.352	2.055
10	100	30	70	0.136	0.656
15	100	40	60	0.076	0.569
20	100	37	63	0.050	0.231
25	100	36	64	0.036	0.116
50	100	37	63	0.014	0.044
75	100	58	42	0.006	0.026
100	100	53	47	0.005	0.021

Given the results of our computational experiments, we conclude that Algorithm 2 is quite effective in solving the multi-job problem where Machine 1 is the primary machine.

5. Conclusions and future research

In this study, we considered a problem in a two-machine re-entrant flow shop in which lot streaming is used for scheduling the single-job and multi-job cases separately. When Machine 1 or Machine 2 is the primary machine on which the third operation is performed, we proved that the single-job problem is polynomial-time solvable. We have also proved that the multi-job problem can be solved optimally when the third operation is performed on Machine 2. However, we have also proved that the multi-job problem is *NP*-hard when Machine 1 is the primary machine, machine so that we have developed a simple heuristic algorithm. To examine the effectiveness of our algorithm, we have developed a global lower bound on the makespan. The results of our computational experiments imply that our heuristic algorithm is quite effective in solving the multi-job problem optimally in reasonably short computational times.

Our study has some limiting assumptions for the problem under study. As in most studies in the lot streaming literature, we assume that the number of sublots of each job lot is known in advance. However, a more realistic case is when the problem's solution has to give its value along with the sublot sizes.

In our study, as in almost all of the previous studies in the literature, we assume that sublots in each job lot should be processed successively for each operation on each machine. i.e., we do not allow the intermingling of the job lots for the multi-job case. We may or may not obtain a better schedule by relaxing this assumption, but it is worth investigating.

Furthermore, in our study, sublot sizes may not be integral, so rounding them to the nearest integer numbers without violating the job-lot size may be needed. However, this approach may not provide the optimal makespan. Thus, this issue is also worthy of investigating.

Through our study for two machine and re-entrant flowshops, we hope that researchers working on scheduling problems will be aware that there is no study other than ours for scheduling with lot streaming in re-entrant manufacturing systems. Our study will open a new direction in the literature of scheduling with lot streaming since it is the only study that applies lot streaming for single- and multi-job cases in the re-entrant flow shops. We hope that our work will form a basis for developing algorithms to solve the scheduling problems in more complex re-entrant manufacturing systems where lot streaming is allowed.

There are several research extensions of this study that are open for future investigation:

- The assumption of knowing the number of sublots in advance could be relaxed, and investigating the problem under consideration without this assumption could be a future study issue.

- The sublot sizes obtained for the no-setup case studied in this paper may not be valid for the setup case The so that problem under consideration can be extended to a case where an issue with attached or detached setup is made before processing a job-lot.

- Our study assumes that sublots in each job-lot should be processed successively for each operation on each machine. i.e., we do not allow the intermingling of the job lots for the multi-job case. Relaxing this the no-intermingling assumption could be another future research issue.

- Our study could also be extended to the flow shops with more than two stages, having one machine at each stage, and a job lot may visit some stages more than once. and hybrid flow shops, which are the flow shops in which at least one of the stages has more than one machine, could also be studied.

- Different measures of performance rather than makespan could also be considered. For instance, the total completion time of sublots and job lots in the single- and multi-job cases, respectively, could be a more relevant performance measure than makespan if the inventory holding costs are minimized.

Appendix A.

Proof of Theorem 1. As illustrated in the network representation given in Figure 1, a lower bound on the makespan $C_{\max}$, which is based on the first sublot, is given as the first sublot's processing time on M_1 plus the total processing time of all sublots on M_2, i.e., $C_{\max} \geq LB_1 = p_1 x_1 + (p_2 + p_3)(x_1 + x_2 + \ldots + x_s)$. Similarly, the total processing time of sublots 1 and 2 plus the total processing time of sublots 2 through s gives another lower bound, i.e., $C_{\max} \geq LB_2 = p_1(x_1 + x_2) + (p_2 + p_3)(x_2 + x_3 + \ldots + x_s)$. Similarly, we can write $C_{\max} \geq LB_i = p_1 \sum_{k=1}^{i} x_k + (p_2 + p_3) \sum_{k=i}^{s} x_k$ for each sublot $i = 2,\ldots,s$. It follows that $C_{\max} \geq \max_{1\leq i \leq s} LB_i = \max_{1\leq i\leq s} \left\{ p_1 \sum_{k=1}^{i} x_k + (p_2 + p_3) \sum_{k=i}^{s} x_k \right\}$. Thus, the linear programming model below can formulate the single-job problem where M_2 is the primary machine.

$$(P) \quad \text{Minimize} \quad C_{\max} \tag{A.1}$$

$$\text{Subject to} \quad C_{\max} \geq p_1 \sum_{k=1}^{i} x_k + (p_2 + p_3) \sum_{k=i}^{s} x_k \quad \text{for } i = 2,\ldots,s \tag{A.2}$$

$$\sum_{i=1}^{s} x_i = U \tag{A.3}$$

$$C_{\max} \geq 0 \tag{A.4}$$

$$x_i \geq 0 \quad \text{for } i = 2,\ldots,s \tag{A.5}$$

Assuming that each of the inequalities in (A.2) is satisfied as equality, i.e., all sublots are critical to determining the makespan $C_{\max}$; we can obtain a feasible solution to problem P. In such a case, both M_1 and M_2 operate without idle time from one sublot to another, and the adjacent pair of sublots must satisfy the following relationship:

$$p_1 \sum_{k=1}^{i} x_k + (p_2 + p_3) \sum_{k=i}^{s} x_k = p_1 \sum_{k=1}^{i-1} x_k + (p_2 + p_3) \sum_{k=i-1}^{s} x_k \quad \text{for } i = 2,\ldots,s \tag{A.6}$$

or equivalently,

$$x_i = x_{i-1}(p_2 + p_3)/p_1 \quad \text{for } i = 2,\ldots,s \tag{A.7}$$

The idle time on M_2 is only before the first sublot, and it equals $p_1 x_1$. Then solving the set of simultaneous equations (A.3) and (A.7) yields

$$x_1 = U/(1 + \alpha + \alpha^2 + \ldots + \alpha^{s-1}), \tag{A.8}$$

$$x_i = \alpha^{i-1} x_1 \quad \text{for } i = 2,\ldots,s \tag{A.9}$$

where $\alpha = (p_2 + p_3)/p_1$.

Using (A.2), (A.3), (A.8) and (A.9), the makespan is obtained as

$$C_{\max} = p_1 x_1 + (p_2 + p_3)\sum_{i=1}^{s} x_i = (p_1 / (1+\alpha++\alpha^2 + \ldots + \alpha^{s-1}) + (p_2 + p_3))\ U\ . \qquad \text{(A.10)}$$

We have shown that the solution given by the theorem is feasible. Note that this solution is the solution for the single-job lot streaming problem in the two-machine flow shop with processing times p_1 and $p_2 + p_3$ on M_1 and M_2, respectively.

Now, to prove the optimality of this feasible solution, the problem P may be re-written as

(P') Maximize $-C_{\max}$ (A.11)

Subject to $-C_{\max} + p_1 \sum_{k=1}^{i} x_k + (p_2 + p_3)\sum_{k=i}^{s} x_k \le 0$ for $i = 2,\ldots,s$ (A.12)

$\sum_{i=1}^{s} x_i = U$ (A.4)

$C_{\max} \ge 0$, $x_i \ge 0$ for $i = 2,\ldots,s$ (A.5)

The dual of (P') is constructed as

(D') Minimize $U\ y_0$ (A.13)

Subject to $p_1 \sum_{k=1}^{s} y_k + (p_2 + p_3)\sum_{k=1}^{i} y_k - y_0 \ge 0$ for $i = 2,\ldots,s$ (A.14)

$-\sum_{i=1}^{s} y_i \ge -1$ (A.15)

$y_i \ge 0$ for $i = 2,\ldots,s$ (A.16)

y_0 unrestricted in sign (A.17)

We can find a feasible solution to problem D' by assuming that all constraints defined in the dual problem D' are satisfied as equalities. It follows that a feasible solution to problem D' is achieved when

$$p_1 \sum_{k=1}^{s} y_k + (p_2 + p_3)\sum_{k=1}^{i} y_k = p_1 \sum_{k=i-1}^{s} y_k + (p_2 + p_3)\sum_{k=1}^{i-1} y_k \text{ for } i = 2,\ldots,s \qquad \text{(A.18)}$$

or equivalently,

$$y_i = y_{i-1} p_1 / (p_2 + p_3) \text{ for } i = 2,\ldots,s\ . \qquad \text{(A.19)}$$

Solving the set of equations in (A.18) and $\sum_{1\le i\le s} y_i = 1$ simultaneously yields

$$y_1 = \alpha^{s-1}(1/(1+\alpha++\alpha^2 + \ldots + \alpha^{s-1}))\ , \qquad \text{(A.20)}$$

$$y_i = \alpha^{1-i} y_1 \text{ for } i = 2,\ldots,s\ , \qquad \text{(A.21)}$$

$$y_0 = p_1 / (1+\alpha++\alpha^2 + \ldots + \alpha^{s-1}) + (p_2 + p_3) \qquad \text{(A.22)}$$

where $\alpha = (p_2 + p_3)/\, p_1$.

The objective function of the dual problem $D1'$ is obtained as

$$U\,y_0 = (p_1/(1+\alpha+\alpha^2+...+\alpha^{s-1})+(p_2+p_3))\,U\,. \tag{A.23}$$

From equations (A.10) and (A.23), it is clear that the problem P' and its dual D' have the same objective function values. Thus, from the duality theory, it follows that equations (A.8)-(A.10) and (A.20)-(A.22) are the optimal solutions to the primal and dual problems, respectively. Therefore, we can conclude that the sublot sizes given in the theorem statement are optimal for the single-job problem where M_2 is the primary machine on which the third operation is performed. ■

Appendix B.

Proof of Theorem 3. For a job sequence π, let

[*j*] = job lot sequenced in the j th position of the sequence π,

$U_{[j]}$ = lot size of job lot [*j*],

$s_{[j]}$ = number of sublots of job lot [*j*],

i = index for sublots ($i=1,...,s_{[j]}$),

$x_{[j],i}$ = size of the i th sublot in job lot [*j*],

k = index for machines ($k=1,2,3$),

$p_{[j],k}$ = item (unit) processing time for the k th operation of job lot [*j*],

$C_{[j],i,k}$ = completion time of the k th operation for the i th sublot of job lot [*j*],

$RI_{[j]}$ = time lag between the start of the first operation on M_1 and the latest start time of the second operation on M_2 for job lot [*j*],

$RO_{[j]}$ = time lag between the completion times of the first and the third operations for job lot [*j*].

The completion time of the first operation for the i th sublot of job lot [*j*] is given by

$$C_{[j],i,1} = C_{[j],s_{[j-1]},1} + \sum_{r=1}^{i} p_{[j],1}x_{[j],r} \text{ for } i=1,2,...,s_{[j]} \tag{B.1}$$

where $C_{[0],s_{[0]},1}=0$. From equation (B.1), the completion time for the first operation of job lot [*j*], which is the completion time of the last sublot in this job on M_1, is obtained as

$$C_{[j],s_{[j]},1} = C_{[j-1]),s_{[j-1]},1} + \sum_{i=1}^{s_{[j]}} p_{[j],1}x_{[j],i} = C_{[j-1],s_{[j-1]},1} + p_{[j],1}\;U_{[j]}\,. \tag{B.2}$$

The completion time of the third operation for the first sublot of job lot [*j*] on M_2 is expressed as

$$C_{[j],1,3} = \max\left\{C_{[j],1,1},\; C_{[j-1],s_{[j-1]},3}\right\} + p_{[j],2}x_{[j],1} + p_{[j],3}x_{[j],1}$$

$$= \max\left\{C_{[j],1,1},\ C_{[j-1],s_{[j-1]},3}\right\} + (p_{[j],2} + p_{[j],3})x_{[j],1}\,. \tag{B.3}$$

Substituting $C_{[j],1,1}$ value from (B.2) into (B.3) yields

$$C_{[j],1,3} = \max\left\{C_{[j-1],s_{[j-1]},1} + p_{[j],1}x_{[j],1},\ C_{[j-1],s_{[j-1]},3}\right\} + (p_{[j],2} + p_{[j],3})x_{[j],1}\,. \tag{B.4}$$

Similarly, we may obtain the completion time of the third operation for the second sublot of job lot [j] on M_2 as

$$C_{[j],2,3} = \max\left\{C_{[j-1],s_{[j-1]},1} + \max_{1\le i\le 2}\left\{\sum_{r=1}^{i} p_{[j],1}x_{[j],r} - \sum_{r=1}^{i-1}(p_{[j],2} + p_{[j],3})x_{[j],r}\right\},\ C_{[j-1],s_{[j-1]},3}\right\}$$
$$+\sum_{i=1}^{2}(p_{[j],2} + p_{[j],3})x_{[j],i} \tag{B.5}$$

Repeating the process yields the completion time for the last operation of job lot [j], which is the completion time of the last sublot of this job lot on M_2,

$$C_{[j],s_{[j]},3} = \max\left\{C_{[j-1],s_{[j-1]},1} + \max_{1\le i\le s_{[j]}}\left\{\sum_{r=1}^{i} p_{[j],1}x_{[j],r} - \sum_{r=1}^{i-1}(p_{[j],2} + p_{[j],3})x_{[j],r}\right\},\right.$$
$$\left. C_{[j-1],s_{[j-1]},3}\right\} + \sum_{i=1}^{s_{[j]}}(p_{[j],2} + p_{[j],3})x_{[j],i}$$
$$= \max\left\{C_{[j-1],s_{[j-1]},1} + \max_{1\le i\le s_{[j]}}\left\{\sum_{r=1}^{i} p_{[j],1}x_{[j],r} - \sum_{r=1}^{i-1}(p_{[j],2} + p_{[j],3})x_{[j],r}\right\},\right.$$
$$\left. C_{[j-1],s_{[j-1]},3}\right\} + (p_{[j],2} + p_{[j],3})U_{[j]} \tag{B.6}$$

By successive application of (B.6) using (B.2), the time to complete the last job lot processed on M_2, $C_{[n],s_{[n]},3}$, is obtained as follows:

$$C_{[n],s_{[n]},3} = C_{\max} = \max_{1\le w\le n}\left\{\sum_{j=1}^{w} RI_{[j]} - \sum_{j=1}^{w-1} RO_{[j]}\right\} + \sum_{j=1}^{n}(p_{[j],2} + p_{[j],3})U_{[j]} \tag{B.7}$$

where

$$RI_{[j]} = \max_{1\le i\le s_{[j]}}\left\{\sum_{r=1}^{i} p_{[j],1}x_{[j],r} - \sum_{r=1}^{i-1}(p_{[j],2} + p_{[j],3})x_{[j],r}\right\}, \tag{B.8}$$

$$RO_{[j]} = \max_{1\le i\le s_{[j]}}\left\{\sum_{r=1}^{i} p_{[j],1}x_{[j],r} - \sum_{r=1}^{i-1}(p_{[j],2} + p_{[j],3})x_{[j],r}\right\} + (p_{[j],2} + p_{[j],3})U_{[j]} - p_{[j],1}U_{[j]}$$
$$= RI_{[j]} + (p_{[j],2} + p_{[j],3} - p_{[j]})U_{[j]} \tag{B.9}$$

Note that the second part, $\sum_{j=1}^{n}(p_{[j],2} + p_{[j],3})U_{[j]}$, in (B.7) giving the makespan value is a constant so that it is enough to minimize the first part, $\max_{1\le w\le n}\left\{\sum_{j=1}^{w} RI_{[j]} - \sum_{j=1}^{w-1} RO_{[j]}\right\}$, which is equivalent to the total idle time on M_2. Note that the first part is similar to Johnson's expression, where the processing times on M_1 and M_2 are replaced by the run-in and run-out delays, respectively. Therefore, job v precedes job z in an optimal schedule of job lots when $\min\{RI_v, RO_z\} \leq \min \{RI_z, RO_v\}$. ■

Appendix C.

Proof of Theorem 4. It is clear that the minimizing the first term in (B.7), $\max_{1\leq i\leq s_{[j]}} \left\{\sum_{r=1}^{i} p_{[j],1}x_{[j],r} - \sum_{r=1}^{i-1}(p_{[j],2}+p_{[j],3})x_{[j],r}\right\}$, minimizes the makespan for a given sequence of job lots. In other words, sublot sizes minimizing the makespan for any arbitrary sequence (hence the optimal sequence) are identical to those sublot sizes which are determined by solving the sublot-sizing problem for each job lot separately. Therefore, the following linear programming model should be solved for every job lot in position [*j*]:

Minimize $Z_{[j]}$ (C.1)

Subject to $Z_{[j]} \geq \sum_{r=1}^{i} p_{[j],1}x_{[j],r} - \sum_{r=1}^{i-1}(p_{[j],2}+p_{[j],3})x_{[j],r}$ for $i = 1, 2, \ldots, s_{[j]}$ (C.2)

$$\sum_{i=1}^{s_{[j]}} x_{[j],i} = U_{[j]} \tag{C.3}$$

$$x_{[j],i} \geq 0 \text{ for } i = 1, 2, \ldots, s_{[j]}. \tag{C.4}$$

The optimal solution to this model is trivial due to its special structure. The minimum value of the objective function $Z_{[j]}$ is achieved when $Z_{[j]} = p_{[j],1}x_{[j],1}$ and the constraints in (C.2) are satisfied as equalities. Thus, the sublot sizes must be

$$x_{[j],i} = x_{[j],i-1}(p_{[j],2}+p_{[j],3})/p_{[j],1} = x_{[j],1}((p_{[j],2}+p_{[j],3})/p_{[j],1})^{i-1} \text{ for } i = 2, \ldots, s_{[j]}. \tag{C.5}$$

Substituting (C.5) into (C.3) yields

$$x_{[j],1} = U_{[j]}/(1+\alpha+\alpha^2+\ldots+\alpha^{s_{[j]}-1}), \tag{C.6}$$

$$x_{[j],i} = \alpha^{i-1}x_{[j],1} \text{ for } i = 2, \ldots, s_{[j]} \tag{C.7}$$

where $\alpha = (p_{[j],2}+p_{[j],3})/p_{[j],1}$.

Note that (C.6) and (C.7) are the same expressions as given in Section 3 for the single-job problem, and substituting (C.6) and (C.7) into (B.8) and (B.9) yields

$$RI_{[j]} = \max_{1\leq i\leq s_{[j]}} \left\{\sum_{r=1}^{i} p_{[j],1}x_{[j],r} - \sum_{r=1}^{i-1}(p_{[j],2}+p_{[j],3})x_{[j],r}\right\} = p_{[j],1}x_{[j],1}$$

$$= p_{[j],1}U_{[j]}/(1+\alpha+\alpha^2+\ldots+\alpha^{s_{[j]}-1}), \tag{C.8}$$

$$RO_{[j]} = RI_{[j]} + (p_{[j],2}+p_{[j],3}-p_{[j]})U_{[j]}. \tag{C.9}$$

Acknowledgements: We would like to thank the editor and anonymous referees for their valuable comments and suggestions that helped us to improve the quality and presentation of this paper.

References

Alfieri, A., Glass, C. A., & van de Velde, S. L. (2012). Lot streaming in a two-machine flow shop with attached setup times. IIE Transactions, 44(8), 695–710. https://doi.org/10.1080/0740817X.2011.649384

Alfieri, A., Zhou, S., Scatamacchia, R., & van de Velde, S. L. (2021). Dynamic programming algorithms and Lagrangian lower bounds for a discrete lot streaming problem in a two-machine flow shop. 4OR, 19(2), 265–288. https://doi.org/10.1007/s10288-020-00449-8

Baker, K. R. (1988). Lot streaming to reduce cycle time in a flow shop, Working Paper, The Amos Tuck School of Business Administration, Dartmouth College, Hanover, NH, USA.

Baker, K. R. (1995). Lot streaming in the two-stage flow shop with setup times. Annals of Operations Research, 57(1), 1–11. https://doi.org/10.1007/BF02099687

Buscher, U. & Shen, L. (2009). An integrated tabu search algorithm for the lot streaming problem in job shops. European Journal of Operational Research, 199(2), 385–399. https://doi.org/10.1016/j.ejor.2008.11.046

Çetinkaya, F. C. (1994). Lot streaming in a two-stage flow shop with set-up, processing and removal times separated. Journal of Operational Research Society, 45(12), 1445–1455. https://doi.org/10.1057/jors.1994.221

Çetinkaya, F. C. (2005). Unit sized transfer batch scheduling in an automated two-machine flow-line cell with one transport agent. International Journal of Advanced Manufacturing Technology, 29(1–2), 178–183. https://doi.org/10.1007/s00170-004-2493-9

Çetinkaya, F. C., & Kayalıgil, M. S. (1992). Unit sized transfer batch scheduling with setup times. Computers & Industrial Engineering, 22(2), 177–183. https://doi.org/10.1016/0360-8352(92)90045-L

Çetinkaya, F. C., & Duman, M. (2010). Lot streaming in a two-machine mixed shop. International Journal of Advanced Manufacturing Technology, 49(9–12), 1161–1173. https://doi.org/10.1007/s00170-009-2473-1

Chang, J. H., & Chiu, H. N. (2005). A comprehensive review of lot streaming. International Journal of Production Research, 43(8), 1515–1536. https://doi.org/10.1080/00207540412331325396

Chen, J., & Steiner, G. (1997). Lot streaming with detached setups in three-machine flow shops. European Journal of Operational Research, 96(3), 591–611. https://doi.org/10.1016/S0377-2217(96)00091-4

Chen, J., & Steiner, G. (1998). Lot streaming with attached setups in three-machine flow shops. IIE Transactions, 30(11), 1075–1084. https://doi.org/10.1023/A:1007563814941

Cheng, M., Mukherjee, N. J., & Sarin, S. C. (2013). A review of lot streaming. International Journal of Production Research, 51(23–24), 7023–7046. https://doi.org/10.1080/00207543.2013.774506

Cheng, M., & Sarin, S. C. (2020). Lot streaming in a two-stage assembly system. Annual Reviews in Control, 50(23–24), 303–316. https://doi.org/10.1016/j.arcontrol.2020.08.004

Cheng, M., Sarin, S.C., & Singh, S. (2016). Two-stage, single-lot, lot streaming problem for a 1+2 hybrid flow shop. Journal of Global Optimization, 66(2), 263–290. https://doi.org/10.1007/s10898-015-0298-z

Davendra, D., Senkerik, R., Zelinka, I., Pluhacek, M., & Bialic-Davendra, M. (2014). Utilising the chaos-induced discrete self organising migrating algorithm to solve the lot-streaming flowshop scheduling problem with setup time. Soft Computing, 18(4), 669–681. https://doi.org/10.1007/s00500-014-1219-7

Defersha, F. M., & Chen, M. (2010). A hybrid genetic algorithm for flowshop lot streaming with setups and variable sublots. International Journal of Production Research, 48(6), 1705–1726. https://doi.org/10.1080/00207543.2011.574952

Defersha, F. M. (2011). A comprehensive mathematical model for hybrid flexible flowshop lot streaming problem. International Journal of Industrial Engineering Computations, 2(2), 283–294. https://doi.org/ 10.5267/j.ijiec.2010.07.006

Defersha, F. M., & Chen, M. (2012a). Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem. The International Journal of Advanced Manufacturing Technology, 62(1–4), 249–265. https://doi.org/10.1007/s00170-011-3798-0

Defersha, F. M., & Chen, M. (2012b). Jobshop lot streaming with routing flexibility, sequence-dependent setups, machine release dates and lag time. International Journal of Production Research, 50(2), 2331–2352. https://doi.org/10.1080/00207543.2011.574952

Feldmann, M., & Biskup, D. (2008). Lot streaming in a multiple product permutation flow shop with intermingling. International Journal of Production Research, 46(1), 197–216. https://doi.org/10.1080/00207540600930065

Glass, C. A., Possani, E. (2011). Lot streaming multiple jobs in a flow shop. International Journal of Production Research, 49(9), 2669–2681. https://doi.org/10.1080/00207543.2010.532935

Gomez-Gasquet, P., Segura-Andres, R., & Andres-Romano, C. (2013). A review of lot streaming in a flow shop environment with makespan criteria. Journal of Industrial Engineering and Management, 6(3), 761–770. http://dx.doi.org/10.3926/jiem.553

Johnson, S. M. (1954). Optimal two and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1(1), 61–68. https://doi.org/10.1002/nav.3800010110

Kim, J. S., Kang, S. H., & Lee, S. M. (1997). Transfer batch scheduling for a two-stage flowshop with identical parallel machines at each stage. Omega-International Journal of Management Science, 25(5), 547–555. https://doi.org/10.1016/S0305-0483(97)00015-7

Lev, V., & Adiri, I. (1984). V-shop scheduling. European Journal of Operational Research, 18(1), 51–56. https://doi.org/10.1016/0377-2217(84)90260-1

Li, J. Q., Tao, X. R., Jia, B. X., Han, Y. Y., Liu, C., Duan, P. Zheng, Z. X., & Sanga, H. Y. (2020). Efficient multi-objective algorithm for the lot-streaming hybrid flowshop with variable sub-lots. Swarm and Evolutionary Computation, 5, Article 100600. https://doi.org/10.1016/j.swevo.2019.100600

Liu, J. (2008). Single-job lot streaming in m-1 two-stage hybrid flowshops. European Journal of Operational Research, 187(3), 1171–1183. https://doi.org/10.1016/j.ejor.2006.06.066

Marimuthu, S., Ponnambalam, S. G., Jawahar, N. (2008). Evolutionary algorithms for scheduling m-machine flow shop with lot streaming. Robotics and Computer-Integrated Manufacturing, 24(1), 125–139. https://doi.org/10.1016/j.rcim.2006.06.007

Marimuthu, S., Ponnambalam, S. G., Jawahar, N. (2009). Threshold accepting and Ant-colony optimization algorithms for scheduling m-machine flow shops with lot streaming. Journal of Materials Processing Technology, 209(2), 1026–1041. https://doi.org/10.1016/j.jmatprotec.2008.03.013

Martin, C. H. (2009). A hybrid genetic algorithm/mathematical programming approach to the multi-family flowshop scheduling problem with lot streaming. Omega, 37(1), 126–137. https://doi.org/10.1016/j.omega.2006.11.002

Meng, T., Pan, Q. K., Li, J. Q., & Sang, H. Y. (2018). An improved migrating birds optimization for an integrated lot-streaming flow shop scheduling problem. Swarm and Evolutionary Computation, 38, 64–78. https://doi.org/10.1016/j.swevo.2017.06.003

Naderi, B, & Yazdani, M. (2015). Modelling and Scheduling Lot Streaming Flexible Flow Lines. Journal of Optimization in Industrial Engineering, 8(18), 61–69. http://www.qjie.ir/article_221.html

Nejati, M., Mahdavi, I., Hassanzadeh, R., & Mahdavi-Amiri, N. (2016). Lot streaming in a two-stage assembly hybrid flow shop scheduling problem with a work shift constraint. Journal of Industrial and Production Engineering, 33(7), 459–471. https://doi.org/10.1080/21681015.2015.1126653

Pan, Q. K., Tasgetiren, M. F., Suganthan, P. N., & Chua, T. J. (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Information sciences, 181(12), 2455–2468. https://doi.org/10.1016/j.ins.2009.12.025

Potts, C. N., & Baker, K. R. (1989). Flow shop scheduling with lot streaming. Operations Research Letters, 8(6), 297–303. https://doi.org/10.1016/0167-6377(89)90013-8

Pranzo, M. (2004). Batch scheduling in a two-machine flow shop with limited buffer and sequence independent setup times and removal times. European Journal of Operational Research, 153(3), 581–592. https://doi.org/10.1016/S0377-2217(03)00264-9

Reither, S. (1966). A system for managing job shop production. Journal of Business, 39(3), 371–393.

Rossit, D., Tohmé, F., Frutos, M., Bard, J & Broz, D. (2016). A non-permutation flowshop scheduling problem with lot streaming: A Mathematical model.

International Journal of Industrial Engineering Computations, 7(3), 507–516. https://doi.org/ 10.5267/j.ijiec.2015.11.004

Salazar-Moya, A., & Garcia, M. V. (2021). Lot Streaming in Different Types of Production Processes: A PRISMA Systematic Review. Designs, 5(4), 67. https://doi.org/10.3390/designs5040067

Sarin, S. C., & Jaiprakash, P. (2007). Flow shop lot streaming. Springer. http://dx.doi.org/10.1007/978-0-387-47688-9

Sarin, S. C., Yao, L., & Trietsch, D. (2011). Single-batch lot streaming in a two-stage assembly system. International Journal of Planning and Scheduling, 1(1–2), 90–108. https://doi.org/10.1504/IJPS.2011.044604

Sriskandarajah, C., & Wagneur, E. (1999). Lot streaming and scheduling multiple products in two-machine no-wait flow shops. IIE Transactions, 31(8), 695–707. https://doi.org/10.1023/A:1007643210135

Şen, A., & Benli, Ö. S. (1999). Lot streaming in open shops. Operations Research Letters, 23(3–5), 135–142. https://doi.org/10.1016/S0167-6377(98)00026-1

Trietsch, D. (1989). Polynomial transfer lot sizing techniques for batch processing on consecutive machines. Working Paper, Naval Postgraduate School, Monterey, California, USA. https://core.ac.uk/download/pdf/36722285.pdf

Trietsch, D., & Baker, K. R. (1993). Basic techniques for lot streaming. Operations Research, 41(6), 1065–1076. https://doi.org/10.1287/opre.41.6.1065

Tseng, C. T., & Liao, C. J. (2008). A discrete particle swarm optimization for lot-streaming flowshop scheduling problem. European Journal of Operational Research, 191(2), 360–373. https://doi.org/10.1016/j.ejor.2007.08.030

Vickson, R. G. (1995). Optimal lot streaming for multiple products in a two-machine flow shop. European Journal of Operational Research, 85(3), 556–575. https://doi.org/10.1016/0377-2217(93)E0366-6

Vickson, R. G., & Alfredson, B. E. (1992). Two-and three-machine flow shop scheduling problems with equal sized transfer batches. International Journal of Production Research, 30(7), 1551–1574. https://doi.org/10.1080/00207549208948107

Wang, M., Sethi. S. P., & Van De Velde, S. L. (1997). Minimizing makespan in a class of re-entrant shops. Operations Research, 45(5), 702–712. https://doi.org/10.1287/opre.45.5.702

Wang, S., (1997). Minimizing makespan in a class of re-entrant shops. Operations Research, 45(5), 702–712. https://doi.org/10.1287/opre.45.5.702

Wang, S., Kurz, M., Mason, S. J., & Rashidi, E. (2019). Two-stage hybrid flow shop batching and lot streaming with variable sublots and sequence-dependent setups. International Journal of Production Research, 57(22), 6893–6907. https://doi.org/10.1080/00207543.2019.1571251

Yang, D. L., & Chern, M. S. (2000). Two-machine flowshop group scheduling problem. Computers & Operations Research, 27(10), 975–985.

https://doi.org/10.1016/S0305-0548(99)00070-2

Yao, L., & Sarin, S. C. (2014). Multiple-lot, lot streaming in a two-stage assembly system. Essays in production, project planning and scheduling (pp. 357–388). Springer. https://doi.org/10.1007/978-1-4614-9056-2_15

Yoon, S. H., & Ventura, J. A. (2002). An application of genetic algorithms to lot-streaming flow shop scheduling. IIE Transactions, 34(9), 779–787. https://doi.org/10.1080/07408170208928911

Zhang, B., Pan, Q. K., Gao, L., Zhang, X. L., Sang, H. Y., & Li, J. Q. (2017). An effective modified migrating birds optimization for hybrid flowshop scheduling problem with lot streaming. Applied Soft Computing, 52, 14–27. https://doi.org/10.1016/j.asoc.2016.12.021

Zhang, W., Liu, J., & Linn, R. J. (2003). Model and heuristics for lot streaming of one job in m-1 hybrid flowshops. International Journal of Operations and Quantitative Management, 9(1), 49–64. http://www.ijoqm.org/v9no1.asp

Zhang, W., Yin, C., Liu, J., & Linn, R. J. (2005). Multi-job lot streaming to minimize the mean completion time in m−1 hybrid flowhops. International Journal of Production Economics, 96(2), 189–200. https://doi.org/10.1016/j.ijpe.2004.04.005