Lot streaming in open shops

Alper Şen[a,*], Ömer S. Benli[b]

[a]*Information and Operations Management Department, Marshall School of Business, University of Southern California, Los Angeles, CA 90089, USA*
[b]*Department of Industrial Engineering, Bilkent University, 06533 Ankara, Turkey*

Abstract

We study lot streaming in shops in which the order the jobs are to be processed is immaterial. For a single job case, the properties of the optimal routing is determined. For the multi job case, two machine shops are analyzed. It is shown that lot streaming will improve makespan only if there is a job with large processing times. We show that there can be at most one such job and derive the optimal sublot sizes and their sequence. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Production/scheduling; Lot streaming; Open shop

1. Introduction

The advances in manufacturing automation and the changing nature of business competitiveness created demand for viable production scheduling systems. Lot streaming is one such avenue of research. Very simply stated, lot streaming is moving some portion of a process batch ahead to begin a downstream operation. Although the specific research in this area started in the mid-1980s [1, 14], its roots can be traced to Group Technology (leading to cell-based manufacturing, resulting in *shorter lead times and reduced work in progress* inventories), just-in-time systems (*lot size of one*) and OPT/synchronous manufacturing (*transfer vs. process batches*). Lot streaming is closely related to batching and lot sizing. Extensive review of

research in this area can be found in Potts and Van Wassenhove [9].

The classical machine scheduling theory envision an operation as an elemental task to be performed. It is assumed that "[t]he processing times of successive operations of a particular job may not be overlapped. A job can be in process on at most one operation at a time" [4]). This assumption may be justified if jobs are monolithic entities. But it is hardly acceptable if we are concerned with production lots consisting of a number of units. The processing time of such a lot is composed of a (usually "detached") setup time plus the product of processing time of a unit and the number of units in the lot. For instance, given that the machine is available, it is not necessary to delay its setup until all items arrive from an upstream machine. Such problems are discussed in the context of deterministic flow shops in Baker [3], Şen et al. [10] and

---

* Corresponding author.

stochastic flow and job shop environments are discussed in Smunt et al. [12]. This paper discusses lot streaming in open shops.

The next section defines the problem. The rest of the paper is organized as follows: The next section defines the problem. Section 3 presents results on single job models. Section 4 discusses lot streaming of multiple jobs in two-machine open shops and states the conditions under which improvements can be achieved.

2. Open shop scheduling and lot streaming

The open shop scheduling model consists of m machines, $M_1, \ldots, M_m$ and n jobs, $J_1, \ldots, J_n$. Each job J_j has m operations, $\{O_{1j}, \ldots, O_{mj}\}$. O_{ij} has a processing of length P_{ij} to be performed on machine M_i. A machine can process at most one job at a time and operations of a job cannot be processed simultaneously. The *routing* for a job is the order of machines that the job visits. If each job J_j is to be processed P_{ij} consecutive time units on machine M_i, the shop is called a *non-preemptive* open shop, otherwise it is a *preemptive* open shop. Define C_{ij} to be the completion time of ith operation of job J_j. The *completion time*, C_j, of a job J_j is the time when all of its m operations are completed, i.e. $C_j = \max_i C_{ij}$. The *makespan* is the maximum of the completion times, i.e. $C_{\max} = \max_j C_j$.

An open shop schedule must satisfy the following two sets of constraints;

- No two jobs can be processed simultaneously on a machine. That is, for each machine M_i and for each pair of jobs (J_j, J_k),

$$\text{either } C_{ij} \geqslant P_{ij} + C_{ik} \text{ or } C_{ik} \geqslant P_{ik} + C_{ij}. \quad (1)$$

- No two machines can process a job simultaneously. That is, for each job J_j and for each pair of machines (M_i, M_l),

$$\text{either } C_{ij} \geqslant P_{ij} + C_{lj} \text{ or } C_{lj} \geqslant P_{lj} + C_{ij} \quad (2)$$

One of the fundamental results in open shop scheduling is the linear time algorithm, proposed by Gonzales and Sahni [7], to minimize makespan in a two machine non-preemptive open shop. We briefly outline the algorithm below.

Denote $a_j = P_{1j}, \; b_j = P_{2j}$.

Algorithm
Step 1: Define $A = \{J_j | a_j \geqslant b_j\}, \; B = \{J_j | a_j < b_j\}$
Step 2: Choose J_r and J_l such that

$$a_r \geqslant \max_{J_j \in A} a_j, \qquad b_l \geqslant \max_{J_j \in B} b_j$$

and let $A' = A - \{J_l, J_r\} \; B' = B - \{J_l, J_r\}$
Step 3:
If $\sum_{j=1}^n a_j - a_l > \sum_{j=1}^n b_j - b_r$,
Construct the schedule (l, B', A', r) on M_1, (r, l, B', A') on M_2, with job J_r having the routing (M_2, M_1), and other jobs (M_1, M_2)
otherwise,
Construct the schedule (B', A', r, l) on M_1, (l, B', A', r) on M_2, with job J_l having the routing (M_2, M_1), and other jobs (M_1, M_2)

The jobs in A' and B' can be ordered arbitrarily.

It can be shown [7] that the algorithm finds a schedule with a makespan,

$$C_{\max} = \max \left\{ \sum_{j=1}^n a_j, \sum_{j=1}^n b_j, \max_j \{a_j + b_j\} \right\}.$$

Since this is a lower bound for the length of any schedule, the algorithm is optimal. Furthermore, Gonzales and Sahni [7] have shown that the problem is NP-*Hard* for $m \geqslant 3$.

One of the interesting and important characteristics of the open shop is the one-to-one correspondence between an m machine n job problem (with processing times P_{ij} of job j on machine i) and the open shop problem with n machines m jobs (with processing times P_{ij} of job i on machine j), where machines are considered as jobs and jobs as machines. This relation is depicted in Fig. 1, with a 2-machine 3-job open shop and its corresponding 3-machine 2-job open shop. As it may be seen, the completion time of jth job on ith machine in the first shop is equal to the completion time of ith job on jth machine in the second shop. Note also that the routing of jth job on the first shop is equivalent to the sequence of jobs on the jth machine on the second shop and vice versa. This correspondence is due to the analogous type (1) and type (2) constraints that both shops are subject to. Optimizing
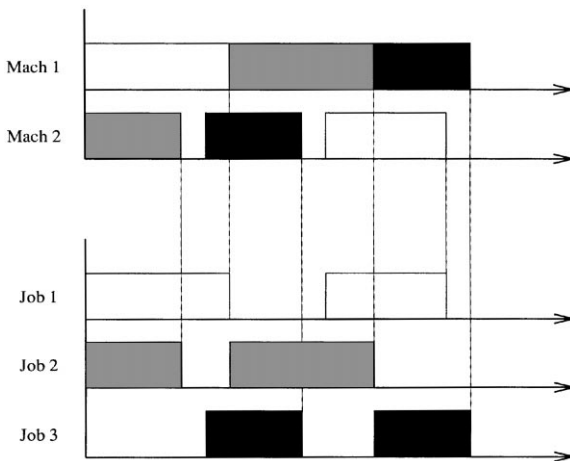
Fig. 1.

makespan in the first shop is equivalent to optimizing makespan in the second shop.

If the jobs are not monolithic and it is possible to split them, then lot streaming can be useful in improving the makespan of an open shop problem. Define U_j to be the number of identical units that forms the job J_j. Let p_{ij} be the unit processing time of jth job on ith machine, i.e. $p_{ij} = (1/U_j)P_{ij}$. Each job J_j can have at most s_j sublots. The problem is to find the routing and the size of these sublots. L_{ijk} represents the size of the kth sublot of job j on machine i.

The sublots are *consistent*, if the sublot sizes are the same at each machine, i.e. $L_{ijk} = L_{jk}$ for each machine i. If we assume that sublots are consistent, then we have similar type (1) and type (2) constraints for each sublot, with $p_{ij}L_{jk}$ being the processing time and C_{ijk} being the completion time of the kth sublot of jth job on machine i. We redefine the completion time as the time when each sublot of each job is completed.

When there is a *single job*, there are two cases to consider. In the first case, all the sublots of the single job may be restricted to follow the same routing, which will be called *single routing* models. In this case, the routing for the job and sizes of the sublots should be optimized. However, an open shop may have further flexibility to allow for different routings for each sublot of the single job, which will be referred to as a *multiple routing* model. In this case we expect to have shorter makespans by optimizing the routing and size of each sublot. If a machine has to be busy from its

first operation to the last one, we have a *continuous work* model.

The problem becomes more complex in *multi job* open shops. In addition to the concepts of single-multiple routings and continuous work, *preempting* a job may improve the makespan. If preemption is allowed, a machine can process sublot(s) of some other job between any two sublots of a job. When the lot streaming is allowed, an open shop is called a *non-preemptive* open shop, if each job J_j has to be processed P_{ij} consecutive time units on machine M_i, over the time when machine M_i is busy. Note however that, in any case, we do not allow for preemption of an individual sublot.

3. Single job models

Since there is only one job, we omit the subscript j in this section. The single job has s sublots and we are to determine the size and routing of these sublots to achieve the minimum makespan. Without loss of generality, let $U \equiv 1$.

3.1. Single routing model

In this case, we assume that the sublots are consistent. There are two decisions to be made: the routing which the job follows and the sizes of the sublots. Clearly, if we are given the routing of the job, the problem turns into a flow shop lot streaming model, for which we can obtain optimal solutions efficiently by linear programming formulations [1].

On the other hand, if we are given the sublot sizes, the problem is only to determine the routing of the job. This model is studied by Steiner and Truscott [13] with the equal sized sublots, i.e. $L_k = 1/s$, $k = 1, \ldots, s$. With the additional restriction that the machines must work continuously ("*continuous work*"), they have shown that in an optimal schedule the job must follow any of the *pyramidal* routings. In a pyramidal routing $R_p = (M_{(1)}, M_{(2)}, \ldots, M_{(m)})$, the job visits the machines with an ascending order of processing times followed by machines with a descending order of processing times, i.e., there is no i such that $p_{(i-1)} > p_{(i)} < p_{(i+1)}$.

Here, we relax the assumption that the sublots must be of equal size and the machines must work continuously. We will show that pyramidal routings are still optimal.

Now, consider the m-machine open shop lot streaming problem. Suppose that sublots are known *a priori* and are $\bar{L} = (\bar{L}_1, \bar{L}_2, \ldots, \bar{L}_s)$. Hence the problem is a classical open shop problem with m machines and s jobs, with processing times,

$$p_{ik} = p_i \bar{L}_k, \quad i=1,\ldots,m \ \ k=1,\ldots,s.$$

But in this specific problem, we also have job $(k+1)$ to follow job k. Therefore, the corresponding s-machine m-job open shop problem is in fact a flow shop problem with processing times,

$$p_{ik} = p_k \bar{L}_i, \ i=1,\ldots,s \ \ k=1,\ldots,m. \quad (3)$$

Observing this relation, we can now use the basic results of the flow shop problem.

When there are two sublots, the corresponding flow shop has two machines. There are two cases to consider: $\bar{L}_1 \geqslant \bar{L}_2$ and $\bar{L}_2 > \bar{L}_1$. When $\bar{L}_1 \geqslant \bar{L}_2$, in the corresponding flow shop, the processing time on the first machine is always greater than the processing time on the second machine for each job, as seen from Eq. (3). By Johnson's Rule, an optimal solution to this problem is LPT sequence for the processing times on machine 2. Thus, the routing in the original open shop, which corresponds to the LPT sequence in the corresponding flow shop, is the routing in which the job visits the machines with a descending order of processing times, i.e. the routing $R_d = (M_{(1)}, M_{(2)}, \ldots, M_{(m)})$ is such that $p_{(i)} \geqslant p_{(i+1)}$ for $i=1,\ldots,m-1$. Similar arguments are valid for the case $\bar{L}_2 > \bar{L}_1$, in which the job visits the machines with an ascending order of processing times, i.e. the routing $R_a = (M_{(1)}, M_{(2)}, \ldots, M_{(m)})$ is such that $p_{(i)} \leqslant p_{(i+1)}$ for $i=1,\ldots,m-1$. Moreover, because of the reversibility of the flow shop lot streaming problem, the two routings give the same makespan. Thus, it is enough to consider only one of these routings. Once the routing is known, the problem is a single job two sublot flow shop lot streaming problem, which can be solved by an LP formulation or by the simple procedure by Baker and Pyke [2].

When there are more than two sublots ($s>2$), we observe the following characteristic of the corresponding s machine flow shop:

$$p_{ik} > p_{il} \Rightarrow p_{hk} > p_{hl},$$

since $p_k \bar{L}_i > p_l \bar{L}_i \Rightarrow p_k \bar{L}_h > p_l \bar{L}_h$.

$$p_{ik} > p_{hk} \Rightarrow p_{il} > p_{hl},$$

since $p_k \bar{L}_i > p_k \bar{L}_h \Rightarrow p_l \bar{L}_i > p_l \bar{L}_h$.

$$i,h \in \{1,\ldots,s\} \ k,l \in \{1,\ldots,m\}.$$

These characteristics are the properties of an ordered flow shop. Smith et al. [11] have shown that the best permutation schedule for this problem is one of the *pyramidal* schedules, i.e. the sequence on any machine $S_p = (J_{(1)}, J_{(2)}, \ldots, J_{(m)})$ is such that there is no k, $1 \leqslant k \leqslant m$ with $p_{(k-1)} > p_{(k)} < p_{(k+1)}$. An immediate result of pyramidal schedules in the corresponding flow shop is the pyramidal routings for the original open shop. Hence, we need to consider one of the 2^{m-1} pyramidal routings.

When the sublots are of equal size, it is easy to see that all the pyramidal routings result in the same makespan. Moreover, it is always possible to ensure continuous work on each machine without increasing the makespan.

3.2. Multiple routing model

In this case, each sublot of the job may have a different routing resulting in shorter makespans. This problem is studied by Glass et al. [6] and the following results are derived.

When the number of sublots available is more than the number of machines, i.e. $s \geqslant m$, optimal sublots are consistent and,

$$L_k = \begin{cases} 1/m & \text{for } k=1,\ldots,m, \\ 0 & \text{for } k=m+1,\ldots,s \end{cases}$$

with sublot k having the routing $(M_k,\ldots,M_m,M_1,\ldots,M_{k-1})$ and achieving a makespan $C_{\max} = \max\{p_1,\ldots,p_m\}$. Note that, in each of the m equal length intervals in the interval $(0, C_{\max})$, each machine processes exactly one of the m sublots and hence there is no overlapping.

When there are two sublots and m machines, optimal sublots are consistent and $L_1 = L_2 = \frac{1}{2}$. The routings of two sublots are found by applying Gonzales and Sahni's [7] algorithm to the corresponding 2-machine m-job problem, generating a makespan, $C_{\max} = \max\{\frac{1}{2}\sum_{i=1}^{m} p_i, \max\{p_1, \ldots, p_m\}\}$.

4. Multi job models

To our knowledge, there is no study of analytical results in the literature on streaming multi jobs in an open shop. Dauzere–Peres and Laserre [5] give an iterative procedure to solve the preemptive open shop, job shop and flow shop problems. The procedure starts with a sequence of sublots on each machine. Given the sequences, optimal sublot sizes are computed. The optimal sublot sizes are now input to a classical scheduling problem where each sublot are assumed to be distinct jobs. The iterative procedure stops when there are no more improvements.

In this section, we study the 2-machine open shop problem. In the first part, we discuss the non-preemptive case where each sublot of a job has the same routing, i.e. "single routing". In the second part, we study the preemptive case where each sublot of a job may have different routings, i.e. "multiple routing". Again, we denote $a_j = P_{1j}$, $b_j = P_{2j}$.

When there are only two machines, Gonzales and Sahni's [7] linear time algorithm finds the optimal schedule with a makespan,

$$C_{\max} = \max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j, \max_j\{a_j + b_j\}\right\}.$$

Clearly, if $\max\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\} \geq \max_j\{a_j + b_j\}$, the makespan cannot be improved by lot streaming. Hence, lot streaming is efficient only if $\max_j\{a_j + b_j\} > \max\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\}$.

For the results of the following subsections, we need the following lemma.

Lemma 1. *There can be at most one job v such that*

$$a_v + b_v > \max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\right\}. \qquad (4)$$

Proof. Suppose that there are two jobs v and l that satisfy,

$$a_v + b_v > \max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\right\},$$

$$a_l + b_l > \max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\right\},$$

which results,

$$a_v + b_v + a_l + b_l > 2\max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\right\}. \qquad (5)$$

On the other hand,

$$\sum_{j=1}^{n} a_j + \sum_{j=1}^{n} b_j \geq a_v + b_v + a_l + b_l, \qquad (6)$$

and

$$2\max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\right\} \geq \sum_{j=1}^{n} a_j + \sum_{j=1}^{n} b_j. \qquad (7)$$

From (5)–(7) we have

$$\max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\right\} > \max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\right\}$$

which is a contradiction. Hence there are no such two jobs v and l. $\square$

4.1. Non-preemptive single routing model

Consider the case, in which there is a job v that satisfies Eq. (4), for otherwise lot streaming will not improve makespan. Without loss of generality, assume that $U_v = 1$. If we consider only job v for streaming, assigning an arbitrary routing (M_1, M_2) or (M_2, M_1) and ignoring the other jobs, the optimal sizes of the s_v sublots are simply the *geometric* sublots given in Potts and Baker [8]. If we take the routing as (M_1, M_2) these sizes are,

$$L_{v1} = \frac{1 - \pi}{1 - \pi^{s_v}},$$

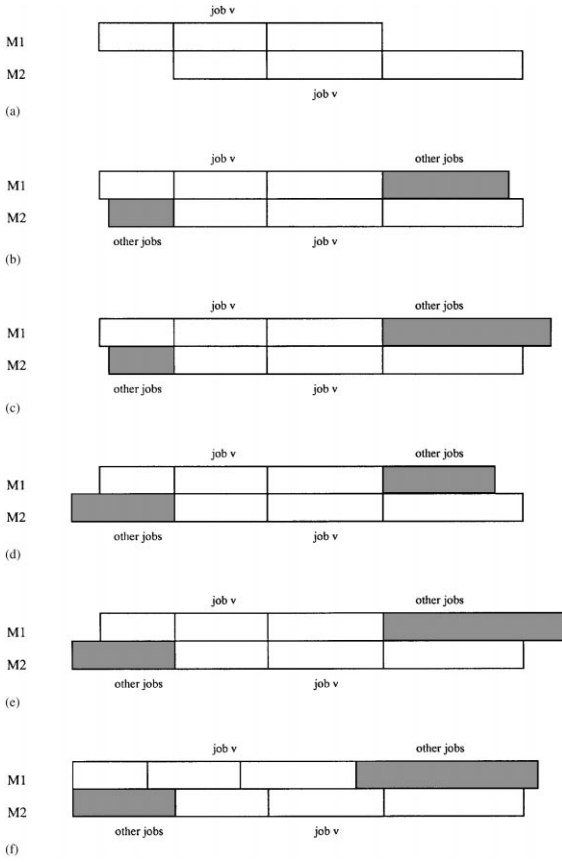$$L_{vk} = \pi L_{k-1}, \quad k = 2, \ldots, s,$$

Fig. 2.

where $\pi \equiv b_v/a_v$. The optimal completion time of the job v is,

$$C_v = a_v L_{v1} + b_v = a_v \frac{1-\pi}{1-\pi^{s_v}} + b_v$$

(see Fig. 2a).

The next step in constructing an optimal schedule is sequencing other jobs after job v on machine M_1 and before job v on machine M_2 (see Fig. 2b). The resulting schedule is optimal if one of the following conditions hold,

$$a_v L_{v1} \geqslant \sum_{j=1}^{n} b_j \quad \text{and} \quad b_v L_{vs_v} \geqslant \sum_{j=1}^{n} a_j$$

which results in a makespan $C_{\max} = C_v$ (Fig. 2b),

$$a_v L_{v1} \geqslant \sum_{j=1}^{n} b_j \quad \text{and} \quad b_v L_{vs_v} < \sum_{j=1}^{n} a_j$$

which results in a makespan $C_{\max} = \sum_{j=1}^{n} a_j$ (Fig. 2c),

$$a_v L_{v1} < \sum_{j=1}^{n} b_j \quad \text{and} \quad b_v L_{vs_v} \geqslant \sum_{j=1}^{n} a_j$$

which results in a makespan $C_{\max} = \sum_{j=1}^{n} b_j$ (Fig. 2d).

However, if

$$a_v L_{v1} < \sum_{j=1}^{n} b_j \quad \text{and} \quad b_v L_{vs_v} < \sum_{j=1}^{n} a_j$$

(Fig. 2e), a left shift of all jobs and sublots on machine M_1 will result in a makespan $C_{\max} = \max\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\}$ (Fig. 2f). Note that this left shift is always possible, since

$$a_v + b_v > \sum_{j=1}^{n} b_j \Rightarrow a_v > \sum_{j=1, j \neq v}^{n} b_j.$$

That is, the processing time of job v on machine M_1 is longer than the total processing of all other jobs on machine M_2 and a left shift on machine M_1 does not create any overlapping.

Since in each case we achieve the makespan,

$$C_{\max} = \max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j, C_v\right\},$$

this construction is an optimal one.

For the continuous case, we can also determine the required number of sublots to have a makespan which achieves the physical limit Y, which is,

$$Y = \max\left\{\sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j\right\} \tag{8}$$

for any two machine shop. Equating this limit to C_v,

$$Y = C_v = a_v \frac{1-\pi}{1-\pi^{s_v}} + b_v$$

which results in

$$s_v = \left\lceil \frac{\log(Y - a_v) - \log(Y - b_v)}{\log b_v - \log a_v} \right\rceil$$

after some manipulation. It can be shown that s_v is a positive integer for all values of a_v and b_v when $a_v \neq b_v$. For $a_v = b_v$,

$$C_v = \frac{a_v}{s_v} + a_v = \frac{(s_v + 1)}{s_v} a_v$$

and the optimal value of s_v is

$$s_v = \left\lceil \frac{a_v}{Y - a_v} \right\rceil .$$

4.2. Preemptive multiple routing model

In this case, each sublot is taken as a separate job. Again we will consider the case when there is a job v that satisfies Eq. (4), for otherwise lot streaming will not improve makespan. We start with the following lemma.

Lemma 2. *For each job l,*

$$\frac{1}{2}(a_l + b_l) \leqslant \max \left\{ \sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j \right\} .$$

Proof.

$$a_l + b_l \leqslant \sum_{j=1}^{n} a_j + \sum_{j=1}^{n} b_j$$

$$\leqslant 2 \max \left\{ \sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j \right\} . \qquad \square$$

We will now show that two sublots of equal size for job v will be sufficient to reduce the makespan to its physical limit, Eq. (8). Take these sublots as distinct jobs $v1$ and $v2$ with processing times $a_{v1} = a_{v2} = \frac{1}{2}a_v$ and $b_{v1} = b_{v2} = \frac{1}{2}b_v$. Then apply the Algorithm of Gonzales and Sahni given in Section 2 to the $n + 1$ jobs. The optimal makespan will be

$$C_{\max} = \max \left\{ \sum_{j=1, j \neq v}^{n} a_j + a_{v1} + a_{v2}, \right.$$

$$\left. \sum_{j=1, j \neq v}^{n} b_j + b_{v1} + b_{v2}, \max_j \{a_j + b_j\} \right\} .$$

Obviously, $\sum_{j=1, j \neq v}^{n} a_j + a_{v1} + a_{v2} = \sum_{j=1}^{n} a_j$ and $\sum_{j=1, j \neq v}^{n} b_j + b_{v1} + b_{v2} = \sum_{j=1}^{n} b_j$. We also have

$$a_l + b_l \leqslant \max \left\{ \sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j \right\}$$

for $l \neq v1$ and $l \neq v2$

from Lemma 1, and,

$$a_l + b_l \leqslant \max \left\{ \sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j \right\}$$

for $l = v1$ or $l = v2$

from Lemma 2. Hence our actual makespan is

$$C_{\max} = \max \left\{ \sum_{j=1}^{n} a_j, \sum_{j=1}^{n} b_j \right\}$$

which is the physical limit.

References

[1] K.R. Baker, Lot streaming to reduce cycle time in a flow shop, Working Paper 203, Amos Tuck School, Dartmouth College, 1988. Presented at the 3rd Annual Conf. on Current Issues in Computer Science and Operations Research, Bilkent University, June 1988.

[2] K.R. Baker, D.F. Pyke, Solution procedures for the lot-streaming problem, Decision Sci. 21 (1990) 475–491.

[3] K.R. Baker, Lot streaming in the two-machine flow shop with setup times, Ann. Oper. Res. 57 (1995) 1–11.

[4] R.W. Conway, W.L. Maxwell, L.W. Miller, Theory of Scheduling, Addison-Wesley, Reading, MA, 1967.

[5] S. Dauzere–Peres, J.B. Lasserre, An Integrated Approach in Production Planning and Scheduling, Lectures Notes in Economics and Mathematical Systems, vol. 411, Springer, Berlin, 1994.

[6] C.A. Glass, J.N.D. Gupta, C.N. Potts, Lot streaming in three-stage production processes, European J. Oper. Res. 75 (1994) 378–394.

[7] T. Gonzales, S. Sahni, Open shop scheduling to minimize finish time, J. Assoc. Comput. Mach. 23 (1976) 665–679.

[8] C.N. Potts, K.R. Baker, Flow shop scheduling with lot streaming, Oper. Res. Lett. 8 (1989) 297–303.

[9] C.N. Potts, L. Van Wassenhove, Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity, J. Oper. Res. Soc. 48 (1992) 395–406.

[10] A. Şen, E. Topaloğlu, Ö. Benli, Optimal streaming of a single job in a two stage flow shop, European J. Oper. Res. 110 (1998) 42–62.

[11] M.L. Smith, S.S. Panwalkar, R.A. Dudek, Flowshop sequencing problem with ordered processing time matrices: a general case, Management Sci. 21 (1975) 544–549.

[12] T.L. Smunt, A.H. Buss, D.H. Kropp, Lot splitting in stochastic flow shop and job shop environments, Decision Sci. 27 (1996) 215–237.

[13] G. Steiner, W.G. Truscott, Batch scheduling to minimize cycle time, flow time and processing cost, IIE Trans. 25 (1993) 90–97.

[14] D. Trietsch, Optimal transfer lots for batch manufacturing: a basic case and extensions, Technical Report NPS-54-87-010, Naval Postgraduate School, Monterey, CA, 1987.