38942

LOT STREAMING IN FLOW SHOPS

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL
ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Engin Topaloğlu
December, 1994

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Ömer S Benli

Assoc. Prof. Ömer S. Benli (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

[illegible]

Assoc. Prof. Mustafa Akgül

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

[illegible]

Assist. Prof. Cemal Akyel

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

[illegible]

Assist. Prof. Selçuk Karabatı

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

[illegible]

Assist. Prof. İhsan Sabuncuoğlu

Approved for the Institute of Engineering and Sciences:

[illegible]

Prof. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

LOT STREAMING IN FLOW SHOPS

Engin Topaloğlu
M.S. in Industrial Engineering
Supervisor: Assoc. Prof. Ömer S. Benli
December, 1994

Lot streaming is permitting partial transfer of processed portions of a job to downstream machines, thus allowing ovelapping operations. The primary motivation is to improve the measures of performance by the quick movement of work in the shop. In this thesis, we study various forms of the lot streaming problem in flow shops to derive the characteristics of optimal solutions. We first analyze single job lot streaming problems, then extend the results to multi-job problems.

When there is a single job, the lot streaming problem is to find the best transfer batch sizes that optimizes the given criterion. We consider three different measures of performance, job, sublot and item completion time criteria. We derive a closed form solution for a special case of job completion time criterion. Under sublot completion time criterion, when the first machine has the largest processing time, we show that partial transfers of equal size are optimal. We propose two polynomial time algorithms for the problem in which only two transfer batches are permitted between each consecutive machines for sublot and item completion time criteria.

In multi-job lot streaming problems, the sequencing and lot streaming decisions must be considered simultaneously. For multi-job problems we investigate the hierarchical application of lot streaming and sequencing decisions.

Key words: Scheduling, Lot Streaming, Flow Shop

ÖZET

AKIŞ TİPİ ÜRETİMDE KAFİLE AKTARMA YÖNTEMLERİ

Engin Topaloğlu
Endüstri Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Ömer S. Benli
Aralık, 1994

Kafile aktarması, bir işin tamamlanmış bölümlerinin daha sonraki makinalara gönderilerek, işlemlerin çakıştırılmasına olanak sağlamaktır. Bu süreçin ana amacı, taşıma kafileleri kullanarak işin atelye içinde hızlı akışını temin etmektir. Bu çalışmada çeşitli tek ve çok işli kafile aktarma problemlerinin optimal çözüm yordamları incelenmiştir.

Tek işli problemler için üç kısıt incelenmiştir: iş, kafile ve parça bitiş zamanlarının enazlanması. İş bitiş zamanının enazlanması amaç fonksiyonu için, özel bir durumda optimal kafile büyüklüklerini veren bir çözüm verilmiştir.

Birinci makinenin en büyük işleme zamanına sahip oldugu durumlarda, eşit büyüklükteki kafilelerin kafile bitiş zamanı amaç fonksiyonunu enazladığı gösterilmiştir Kafile ve parça bitiş zamanlarının enazlanması amaç fonksiyonlarında, yalnız iki kafile kullanımının öngörüldüğü durumlar için polinom zamanlı iki algoritma verilmektedir.

Çok işli problemlerde sıralama ve kafile büyüklüğü kararlarının aynı anda verilmesi gereklidir. Kafile aktarma ve sıralama kararlarının ard arda verilebileceğinin öngörüldüğü ikiden çok makinalı bir problem tipi içinde Johnson Kuralına benzer bir algoritma önerilmiştir.

Anahtar sözcükler: Çizelgeleme, Kafile Aktarma, Akıcı Atelye

To my parents

ACKNOWLEDGEMENT

I am indebted to Assoc. Prof. Ömer S. Benli for suggesting the subject and for his inspiring guidance and encouragement during this study.

I wish to acknowledge my thanks to Assist. Prof. Cemal Akyel for his helpful suggestions and criticism from which I have greatly benefited.

I am also indebted to Assoc. Prof. Mustafa Akgül, Assist. Prof. Selcuk Karabatı and Assist. Prof. İhsan Sabuncuoğlu for showing keen interest to the subject matter and accepting to read and review this thesis.

I would like to thank Assoc. Prof. Barbaros Ç. Tansel and Ferda Can Çetinkaya for their valuable comments on the subject matter.

I am very much grateful to Alper Şen for his extensive cooperation. I would also like to thank my classmates Abdullah Daşçı, Abdullah Çömlekçi and Aydın Selçuk for their valuable discussions, friendship and patience during this study.

Last but not the least, I would like to express special thanks to my family for their great patience in the face of near inexcusable neglect.

Contents

List of Figures

Chapter 1

Introduction

The operations scheduling, as one of the basic areas of production and inventory management, has attracted the attention of many researchers, since the pioneering paper of Johnson [15] on the two machine flow shop scheduling problem. Since then the operations research literature witnessed the rapid development of scheduling research. Now the operations research literature is satiated with innumerable papers, but the publications describing the practical applications of scheduling research results remained relatively scarce. This is of course counter-intuitive to the spirit of operations research. This fact is the result of lack of the instances that the restrictions and the assumptions of the scheduling research results hold. As pointed out by Dudek *et al.* [12] for the flow shop scheduling research,

"We have never been approached by anyone claiming to have a need for solving a problem having the characteristics assumed by most flow shop researchers."

They, then question the existence of the problem, along with flexibility and effectiveness of the algorithms on hand under relaxed situations. In fact, contrary to the definitions of the flow shop in the classical scheduling theory, the flow shops may actually require conditions that are entirely apart from the ones presupposed by the scheduling researchers.

In classical scheduling theory the jobs are assumed to be indivisible single monolithic entities. The fact that a job might be of many identical elements is disregarded, or it is implicitly assumed that partial transfer of completed parts in between the stages of the shop is not possible. However, this is rarely the case. In general, the scheduling problems that arise in the batch production systems, the entities called jobs contain more than a single item.

Basically, *the lot streaming is permitting partial transfer of completed portions of the job to the downstream machines to allow overlapping operations.* The essence of the idea is the use of transfer batches in between the stages of the workshop to increase bottleneck utilization by the quick movement of work over the workshop. Lot streaming can be seen as *splitting* where each split part is considered as a different job. However, *preemption*, which is a mean to adjust the priorities of the jobs, is different from lot streaming. In preemption, the penalty associated with preempting a job is the additional setup incurred when the processing of the preempted job is restarted. Whereas no setup is required before the processing of a lot, since the previous lot contains identical items.

Although the *Manufacturing Resource Planning* (MRPII) systems, especially in the batch production environments, disregard the lot streaming issue, the lot streaming concept is likely to be practiced in real settings, since in the existence of capacity problems, it is quite unreasonable to wait for the entire lot to finish its processing in the current stage, while the downstream machines are idle. In our context, lot is a predetermined medium range production quantity, such as weekly MRP order releases, whereas *sublots* or the so called *transfer batches* are the production quantities determined at operational level.

The development of the lot streaming research is in parallel with the rising awareness of the importance of lead time reduction techniques. In *Just In Time* (JIT) philosophy this is attained through the concepts of lot size of one, zero set-up time. The advantage of using large batches in production is totally eliminated when there is no setup. Therefore the use of small lots cut the manufacturing lead times, and hence increase the effectiveness [7]. In group technology, the same goal is attained through reduced setups, work in process inventory and more efficient specialized material handling equipments.

Importance of overlapping operations is also stressed in *Optimized Production Technology* (OPT) [17, 33], also referred to as *Synchronous Manufacturing* [30]. One of the eight principles of the OPT approach emphasizes the use of transfer batches different from the process batches. From the setup cost point of view large process batches are more attractive, but this does not impose any restriction on the transfer batch size. While avoiding the heavy setup costs by using large batches, the inventory carrying cost can be leveled by the use of small transfer batches. The use of transfer batches increases the utilization of the bottleneck machines, which in turn implies the reduction in the flow time [7].

Until recently, there have been relatively little work on lot streaming. The first known research emphasizing the importance of overlapping operations is due to Szendrovitz [25]. He presents a model to find the economic production quantity in a multi-stage production environment. He assumes that a fixed lot size is manufactured through a fixed sequence of manufacturing operations with a single setup before each operation. He allows the transportation of *sub-batches* to have an overlap between the operations so as to reduce the manufacturing cycle time. Each batch is of an individual unit. He pays no special attention to the optimization of any scheduling criteria, other than pointing out the substantial reduction in manufacturing cycle time. Goyal [14] studies the Szendrovits's model. He tries to find the best sublot size that minimizes the setup plus costs inventory holding costs. Hence, Goyal is after an optimal lot streaming strategy.

Moily [19] calls lot streaming as the *component lot-splitting*, in which the lot size of a component item covers only a fraction of its parent items lot size in multi-stage manufacturing environment. He considers the number of sublots as a decision variable and optimizes the total inventory holding cost plus the order costs with respect to number of sublots and the order size of end item.

1.1 The Problem Definition

A *scheduling problem* can be defined as the problem of allocating available times of *resources* to a number of *tasks* optimally. The resources are *renewable*, in the sense that, they cannot be depleted or are capable of being replaced immediately during the planning period. The resources are usually production related such as machines, material handling equipments but can be other entities such as trucks, teachers, doctors. The definition of task may refer to components, classes, truck loads basically any of activities that require allocation of some renewable resource for a certain period of time. We use the terms machines & jobs referring to resources and tasks, respectively.

In the classical scheduling problem, we basically consider m machines to be scheduled to process n jobs. We denote machines with index i, $i = 1, \ldots, m$ and jobs with index j, $j = 1, \ldots, n$. We assume that a machine can process at most one job at a time, and a job can be processed on only one machine at a time.

As defined by Conway *et al.* [11] an *operation* is an elementary task to be performed. Each (i, j) pair may correspond to a number of operations. But, here we assume that each pair denotes just a single operation and hence an operation is uniquely determined by index (i, j). There might be some partial or full precedence relationships among the operations of a job. The ordered set of relations belonging to a job is called *routing*. The processing time, which is denoted by P_{ij}, is the amount of time needed to perform the operation (i, j).

We assume that each job consists of U_j identical units to be processed on all machines. The processing time of a single unit of job j on machine i is denoted by p_{ij}, where $p_{ij} = P_{ij}/U_j$. For each job and for each stage the number of transfer batches (sublots), s_{ij}, are given. We assume that $1 < s_{ij} < U_j$ for some (i, j) otherwise the lot streaming aspect of the problem will be trivial. The transfer batches of job j between the machines i and $i + 1$ are denoted by $L_{ij1}, L_{ij2}, \ldots, L_{ijs_{ij}}$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. A schedule is a set of all pairs, (S_{ijk}, C_{ijk}), indicating start and end times of the operation of a sublot

(i, j, k). The objective is to find the sequence of jobs and the sizes of transfer batches so as to minimize the given criterion. A feasible schedule is a schedule that allows no overlaps among the pairs (S_{ijk}, C_{ijk}) of machine i, and satisfies possibly, some other restrictions imposed by the shop and job characteristics. Furthermore we make following assumptions.

- The machines are available continuously in the scheduling horizon, that is, the machines are free of breakdowns, and they need no maintenance.
- The machines of the shop are distinct, therefore an operation can be performed by only one machine.
- All the jobs are available at time zero.
- The shop is deterministic, that is all the processing times are known with certainty.
- The transportation times between the machines are negligible.

We mainly consider two optimization criteria, *mean flow time* and *makespan*. The flow time is the time elapsed between the arrival and the completion time of a job. We might also be interested in the flow times of sublots. Since we assume that all the jobs are available at time zero, the flow time of a job is equal to its completion time. As there are more than one job and more than one sublot, the weighted sum of flow times is used as an aggregate measure.

The makespan is defined as the time required to complete the processing requirements of all jobs in the shop, or basically, the maximum flow time of a job or sublot. Minimization of makespan is appropriate if all the U_j units belong to a single customer order. In classical scheduling theory the makespan criterion is known to be one of the simplest criteria, in the sense that, it leads to easier problem formulations . But, it is equally important since an improvement in the makespan, although not necessarily, may imply the improvement in other performance measures.

We consider the lot streaming problems in *flow shop*. The flow shop is the simplest multi-stage shop environment. In flow shop the routing of the jobs are unidirectional, that is, each job must visit the machines in a single given order. Although it is the most restrictive and the easiest one, polynomial-time optimization algorithms are not available for the problems containing more than two machines, other than implicit enumeration of feasible alternatives. Under the makespan criterion two machine flow shop problem is solvable in polynomial time. But the problems beyond two machines is shown to be $\mathcal{NP}-$*complete* [23]. For the mean flow time criterion, however, even two machine problem is not within the boundary of the efficiently solvable problems [23].

When we analyze the problem, we use the following additional notation.

- Π denotes the sequence of jobs or sublots.
- $\Pi(j)$ denotes the j th job in the sequence Π.
- $F(L)$ represents the objective function of the lot streaming problem as a function of the transfer batch sizes $L = [L_{111}, \dots, L_{mns_{mn}}]$.

1.2 Assumptions

Devising exact solution schemes to the scheduling and streaming problems with all the attributes defined above is extremely difficult, if not impossible. The pure multi-machine scheduling problems can be solved efficiently, up to just two machines. Therefore, by taking into account the additional complexity introduced by including the lot streaming problem, we impose some restrictive assumptions on the problems, to be able investigate some well solvable cases.

1.2.1 Continuous vs. Discrete Sublots

In general, a lot streaming problem may arise in any setting where a number of tasks or activities are required to be performed on a number of entities sequentially. But clearly the largest domain, or the potential customer of the streaming problems is the discrete parts manufacturing industry, where all the sublots contain discrete number of units. But we assume that the job to be split into lots is *continuous*, i.e. *infinitely divisible*. Therefore, the sublot sizes might be any real value. This is not a very restrictive assumption if the number of units in the job, U_j, is reasonably large.

We have three main reasons for attacking the continuous sublot version. First, under integral sublots requirement, the formulation of lot streaming problem usually results in integer programming models, hence the discrete sublots problem is expected to be harder. Second, we can obtain reasonably good integral solutions by rounding the continuous sublot solution, especially when the number of items within a job is large. Finally, we may increase our understanding of lot streaming problem with integral sublots, by investigating the characterization of the continuous sublot solutions. These characteristics can later be used to devise heuristics to the discrete sublots problem.

1.2.2 Batch vs. Item Availability

While considering the lot streaming problem, one must pay attention to the two cases arising from the assumptions of the shop. An item is said to be *batch available* if that item cannot be transferred to the downstream machines or to the other sublots, before all other items contained in the same sublot is completed on the current machine. The batch availability assumption is applicable for the machines where the items are produced in batches and it is not possible or practical to remove the item from the batch, until the whole batch completes its processing. The heat treatment and batch painting operations are of this kind. Since the batch availability assumption is a restriction, it may result in sub-optimal solutions, as it will be shown in the next section.

If an item is *item available*, it can be removed from its transfer batch as soon as it completes its processing on the current machine. But, this does not mean that each item is transferred to the next machine as soon as it becomes available. There is still a limit on the number of transfer batches, s_{ij}, to transfer the items from the machine i to machine $i+1$. The item availability assumption is suitable for the cases where an item is dependent to the items in its batch only when the transfers take place.

Under batch availability assumption the completion time of an item on machine i is equal to the completion time time of last item in the sublot it belongs. Whereas under item availability assumption the completion time of an item on the current machine is the time it completes its processing on that machine.

Obviously, the batch and item availability cases may appear together in practice, but for the sake of simplicity we consider the problems where the items are all batch available or all item available in every stage. Also we restrict our attention primarily to the batch availability case, because of its simplicity which results linear type of constraints in the mathematical model.

1.2.3 Consistent vs. Variable Sublots

Under *consistency* assumption, the number of items in a sublot remains fixed across the machines. Hence, $L_{ijk} = L_{jk}$ indicating the machine independent sublot sizes. The consistency assumption is realistic in the settings where frequent changes in the sublot sizes are very costly or, it is difficult to track the movement and contents of the sublots in the shop. Under this assumption the items are by definition batch available.

In *variable sublots* assumption the size of the transfer batches is allowed to change in between the stages of the shop. Allowing variability in the sublot sizes may improve the objective function value. There might also be settings where the variability of sublots arises naturally. For example, consider the shop where the material handling equipment availability is different between the pairs of consecutive machines, so that we are allowed to use more transfer batches in between some stages of the shop. Clearly, this will necessarily result in variable sublots, since it is hardly justifiable not to use existing handling equipment, if the shop is operating around its full capacity. Under consistency assumption, $s_{ij} = s_j$, the number of sublots allowed in between each consecutive stage must be the same for the job j.

To make the discussion more transparent, consider the following example which is a modified version of one given in [21]. A single job of $U = 90$ identical units is to be processed on four machines with processing times $p_1 = 6$, $p_2 = 6$, $p_3 = 90$, $p_4 = 18$. Only two transfer batches are permitted in between the consecutive machines. If streaming is not allowed the makespan, the time complete the whole batch is the number of items times the sum of processing times,

$$C_{max} = U \sum_{i=1}^{m} p_{ij} = 90 \times 120 = 10,800$$

If we allow two consistent transfer batches, the makespan reduces to 9,270, since some of the operations can be overlapped (Figure 1.1). If we also allow variable sublots under batch availability assumption, we are able to decrease the makespan further to 9,180. Notice that on the second machine, the first sublot waits the completion of the second sublot. The reason is that, the first sublot to be send to the next machine, contains items from the second sublot coming from the previous machine. Since the items are batch available, in order to remove one item from the second batch, it is necessary to wait the completion of the entire sublot.

Finally, in the most relaxed case where the sublots are variable under item availability assumption, we can obtain an optimal makespan of length 8,490.

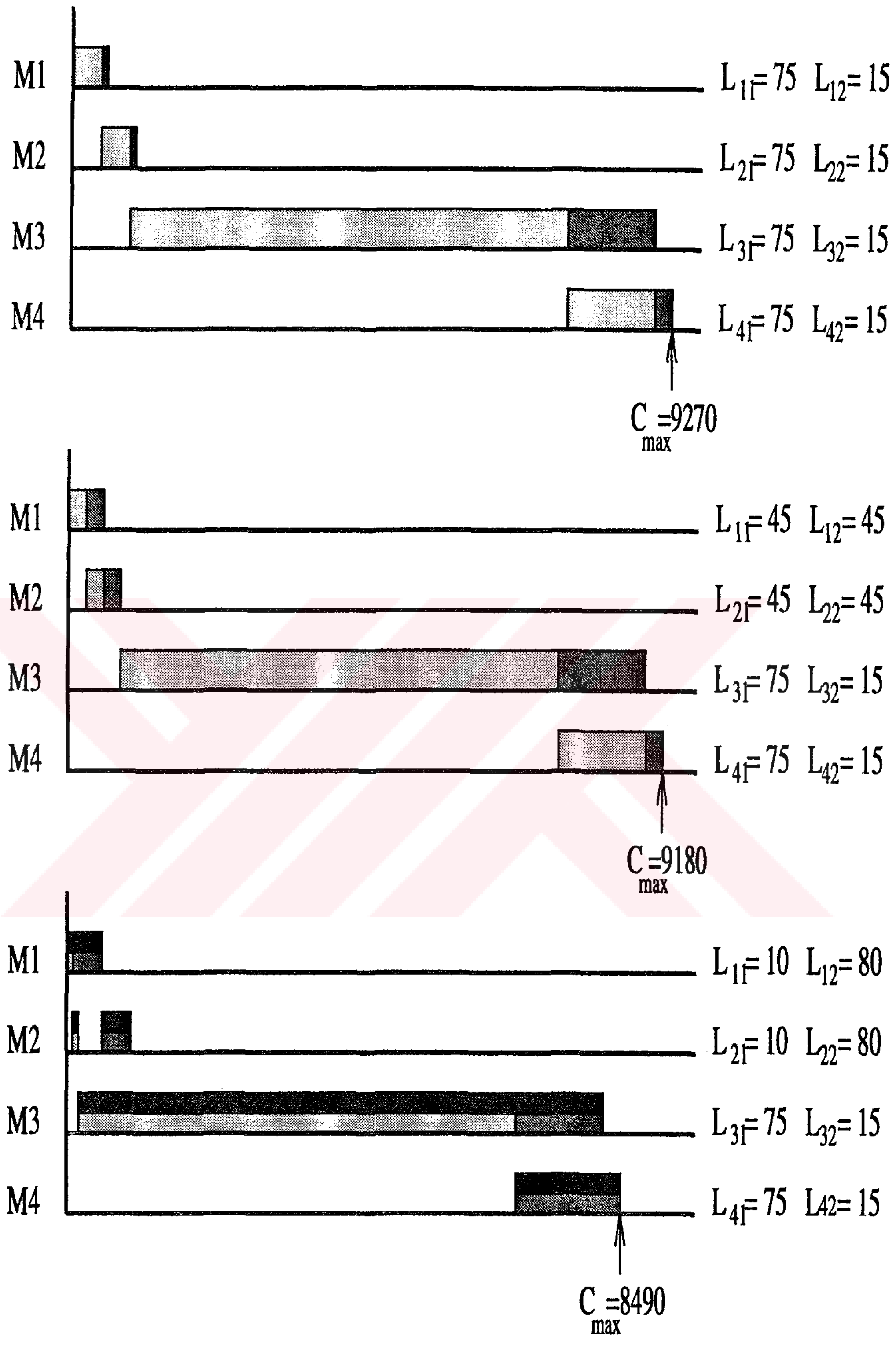

Figure 1.1: Suboptimality of consistent sublots in a 4-machine shop

1.3 Outline of the Thesis

The main purpose of this thesis is to analyze the various lot streaming problems to derive the basic solution characteristics and identify the well solvable cases.

The outline of the thesis is as follows. In Chapter 2 we discuss single job lot streaming problems. We give results pertaining to three measures of the performance: sublot, item, and job completion time criteria. In section 2.2 job completion time criterion is discussed. We derive an exact solution scheme for a special case of m-machine s-sublot problem.

For sublot completion time criterion, we show that when the first machine has the largest processing time, the equal sublots are optimal, in Section 2.3.2. In Section 2.3.3, we give two $O(m^2)$ algorithms for the two sublot problem along with an experimental study for the comparison of the algorithms. An example is presented in Section 2.3.4 to show that the consistent sublot solutions are suboptimal even for two machines.

In Section 2.4 we extend the results of sublot completion time criterion to item completion time criterion. We show that the two sublot problem is solvable in $O(m^2)$ time. The basic results pertaining to two machine problem is also reviewed.

Chapter 3 is devoted to multi-job lot streaming problems. In section 3.2 we give alternative derivation of Vickson's result [31] on two machine lot streaming problem. We present a three machine example in Section 3.3 in which the second machine is dominated, to show that the lot streaming problem is not independent of the sequencing problem. We also discuss the hierarchical application of streaming and sequencing decisions. For a special case of this problem, we show that the sequencing problem can be solved by the Johnson's algorithm [15].

The conclusion and suggestions for further research is presented in the last chapter.

Chapter 2

Single Job Lot Streaming Problems

As discussed in Potts and van Wassenhove [22], three different criteria can be considered depending upon when and how the items are withdrawn from the last machine. When an item cannot be withdrawn from the last machine until the entire job is completed, the criterion is referred to as the *job completion time*. This criterion treats whole U units as a single job, and equivalent to the makespan criterion of classical scheduling theory.

If an item can leave the last machine only when the sublot to which the item belongs completes processing, the completion time of the item is assumed to be the completion time of the last item in its sublot. This criterion is called the minimization of *sublot completion time* and represents a limited delivery capacity from the last machine.

Finally, an item can be delivered as soon as it finishes processing on the last machine. This criterion is called the *item completion time*. In effect, infinite number of transfer batches (sublots) are assumed for the last machine.

The next section defines the general problem, presents the formulation of the general lot streaming problem. The results pertaining to job completion time are presented in Section 2.2. In Section 2.3, optimal results are derived for the criterion of minimizing the sublot completion time. Extensions to the item completion time model are given in Section 2.4.

2.1 Problem

In this study, our treatment of lot streaming problem will be limited to the problems under consistency and infinite divisibility assumptions. But first we give the general formulation of the single job problem.

The lot streaming problem is investigated under consistency, batch availability and continuity assumptions by Baker [4], Potts & Baker [21], Trietsch & Baker [29], but the general case was not addressed. A general formulation of this problem was first given by Benli [6] in the form of mixed integer program. The formulation is able to handle the m-machine s-sublot problem under variable sublots and item availability assumptions. He also models a more realistic case where the number of transfer batches between the machines can vary. Using this formulation, Benli draws some of the well known results, e.g. geometric sublots, characterization of no wait sublots problem etc., by incorporating the restrictions imposed by consistency and other assumptions.

He formulates the lot streaming problem as a multi-stage periodic review model in which the period lengths are allowed to vary. He refers to the output of stage i as item i. Thus, a unit of item $(i-1)$, $i = 1, \ldots, m$ is required in stage i to process a unit of item i, where the item 0 is the raw material and item m is the end product. The periods are defined by the time points at which a transfer occurs between the consecutive machines. Therefore, if s_i transfers are permitted between the stages i and $i+1$, the formulation contains $h = \sum_{i=1}^{m} s_i$ time points where the transfers between the consecutive machines are allowed. If these points are represented by T_t, $t = 1, \ldots, h$, each period of the problem is the intervals $[T_{t-1}, T_t]$. Following variables are used in the formulation,

$X_{i,t}$: Number of items produced at stage i in the period $[T_{t-1}, T_t]$.

$L_{i,t}$: Number of items transferred from stage i to stage $i+1$ in the period $[T_{t-1}, T_t]$.

$I_{i,t}$: Input inventory of item i available at time point T_t to produce item $i+1$.

$O_{i,t}$: Number of items waiting to be transferred to the next machine at time T_t (the output inventory).

$$Y_{i,t} : \begin{cases} 1 & \text{if a transfer occurs between the stages } i \text{ and } i+1 \text{ in period } t. \\ 0 & \text{otherwise} \end{cases}$$

In stage i, the production cannot start before time point T_i, since raw material (item 0) cannot be processed to be used in production of the item i before $i-1$ transfers take place. With a similar reasoning no item can be produced after time period T_t, $t > h-m+i$ since the output cannot be transferred to the last stage using the remaining transfers.

The general formulation of the lot streaming problem is given in the next page. The first two constraint sets maintain the input and output material balance, respectively. The third constraint set specifies the production capacity of the periods. The fourth constraint limits the number of transfers in between the stages i and $i+1$. The fifth constraint set specifies the allowable transfers. The sixth and seventh constraint sets force entire job to be processed. The formulation contains $m(5h-5m-2)+h+1$ variables $m(h-m+1)$ of which are binary variables. There are $4h-3m+1$ constraints excluding the nonnegativity requirements.

$$
\begin{array}{lll}
\min \quad F(L) & & \\
\text{subject to} & & \\
\quad I_{i,t-1} + L_{i-1,t-1} - I_{it} - X_{it} = 0 & i = 1, \ldots, m & t = i, \ldots, h-m+i \\
\quad O_{i,t-1} + X_{it} - O_{it} - L_{it} = 0 & i = 1, \ldots, m & t = i, \ldots, h-m+i \\
\quad p_i X_{it} - (T_t - T_{t-1}) \leq 0 & i = 1, \ldots, m & t = i, \ldots, h-m+i \\
\quad \sum_{t=i}^{h-m+i} Y_{it} \leq s_i, & i = 1, \ldots, m & \\
\quad L_{it} - Y_{it} \leq 0 & i = 1, \ldots, m & t = i, \ldots, h-m+i \\
\quad \sum_{t=m}^{h} L_{mt} = 1 & & \\
\quad L_{0,0} = 1 & & \\
\quad T_0 = I_{i,i-1} = I_{i,h-m+i} = 0 & i = 1, \ldots, m & \\
\quad O_{i,i-1} = O_{i,h-m+i} = 0 & i = 1, \ldots, m & \\
\quad T_t, X_{it}, I_{it}, O_{it}, L_{it} \geq 0 & i = 1, \ldots, m & t = i, \ldots, h-m+i
\end{array}
\tag{2.1}
$$

The objective function is,

$$
F(L) = \begin{cases} T_h & \text{for the job completion time criterion.} \\ \sum_{t=m}^{h-m} L_{it} T_{it} & \text{for the sublot completion time.} \\ \sum_{t=m}^{h-m} L_{it}\{T_{it} - \frac{p_m L_{it}}{2}\} & \text{for the item completion time criterion} \end{cases}
$$

In job completion time minimization, the completion time of the last sublot on the last machine is minimized. In sublot completion time criterion the weighted sum of the completion times of sublots is minimized. The weights are basically the sizes of the sublots. If the criterion is the item completion time, the average completion time of each sublot is weighed by the number of items in the sublot. The formulations specific to the cases under consistency assumption and fixed number of sublots are given in the related sections.

2.2 Job Completion Time Criterion

In this section, the optimization criterion is the minimization of completion time of the last item in the shop or the so called minimization of makespan. Under consistency and infinite divisibility assumptions, the following formulation of single job m-machine flow shop lot streaming problem given independently by Trietsch [28] and Baker [4],

$$
\begin{array}{lll}
\min\ F(L) = C_{ms} & & \\
\text{subject to} & \sum_{k=1}^{s} L_k = U & \\
 & C_{11} - p_1\ L_1 \geq 0 & \\
 & C_{ik} - C_{i,k-1} - p_i\ L_k \geq 0, & i = 1,\ldots,m,\ \ k = 2,\ldots,s \\
 & C_{ik} - C_{i-1,k} - p_i\ L_k \geq 0, & i = 1,\ldots,m,\ \ k = 2,\ldots,s \\
 & C_{ik} \geq 0, & i = 1,\ldots,m,\ \ k = 2,\ldots,s \\
 & L_k \geq 0, & k = 1,\ldots,s
\end{array}
$$

where C_{ik} denotes the completion time of sublot k on machine i, L_k denotes the size of sublot k, and U denotes the number of items in the given job. The objective function value denoted by $F(L)$, is the function of the sublot sizes. Without loss of generality, we let $U = 1$.

The first constraint forces the whole job to be processed. The second and third constraint sets prevent the operations of two consecutive sublots to overlap on a machine. The fourth constraint prevents the overlaps between the operations of a sublot on different machines. In the objective function, the completion time of the last sublot is minimized. Since all the variables are continuous and the objective function and the constraints are linear, this problem is solvable in polynomial time.

2.2.1 General Results

Single job m machine lot streaming problem under consistency assumption is closely related with ordered flow shops in which both machines and jobs are ordered. Given any instance of sublot sizes L_k, $k = 1 \dots s$, the resulting problem is an ordered flow shop problem. This result follows from two facts. First, for every sublot the order of machines from the largest processing time to the smallest processing time is the same. Second, for each machine the ordering of sublots from the ones requiring the greatest processing time to the ones requiring the smallest processing time are the same. Therefore the following results cited by Smith *et al.* [24] highlight some important characteristics of lot streaming problem.

- If the first machine has the greatest processing time then the best consistent sublot sizes are in non-increasing order ($L_k \geq L_{k+1}$).
- If the last machine has the greatest processing time then the best consistent sublot sizes are in non-decreasing order ($L_k \leq L_{k+1}$).
- If the machine with the greatest processing time happens to be one of the middle machines, then the sublot sizes are in pyramidal order, that is, a subset of the sublots are arranged in increasing order of sizes, followed by the remaining sublots in decreasing order.

The proofs of the arguments are straightforward, based on the contradiction using pairwise interchange argument, as shown in Smith *et al.* [24].

Trietsch & Baker [29] present an extensive survey of the basic results of lot streaming under job completion time criterion.

2.2.2 Two-Sublot Problem

A comprehensive treatment of two-sublot problem under makespan criterion is given by Baker & Pyke [5], where an $O(m^2)$ algorithm is developed, by utilizing the bottleneck machine and the critical path concepts. In addition, several heuristics are devised to attack m-machine s consistent sublot problem. They also show that in the optimal solution there are two bottleneck machines. For the same problem, Williams and Tüfekçi [34] presents an algorithm with $O(m^2)$ complexity, and several heuristics, using the network optimization techniques.

2.2.3 Two-Machine Problem

Trietsch [28] and Baker [4] show that for the 2-machine problem the optimal consistent sublots are in geometric pattern (Figure 2.1),

$$L_k = \pi^{k-1}/(\sum_{l=0}^{s-1}\pi^l) \tag{2.2}$$

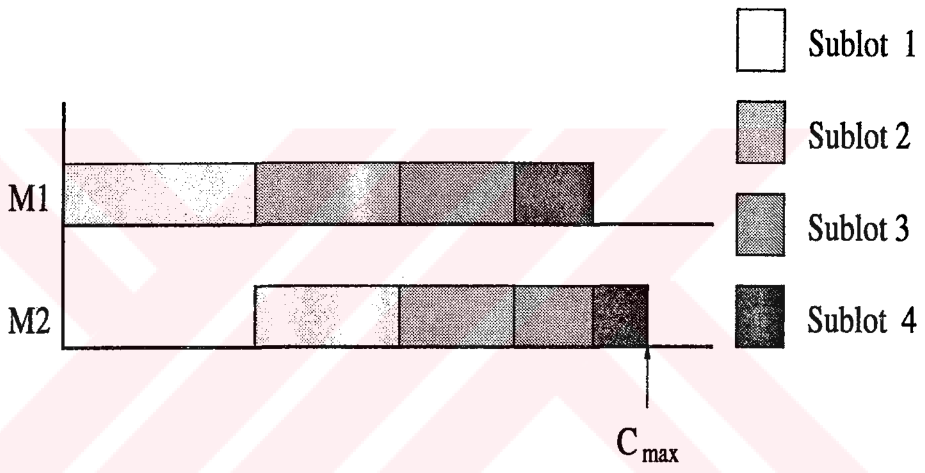

Figure 2.1: 2-machine flow shop with geometric sublots

where L_k is the fraction of k^{th} sublot ($U = 1$), s is the maximum number of transfer batches permitted and π is the ratio of the processing time of second machine to the first ($\pi = p_2/p_1$).

Trietsch [28] studies two machine problem under limited material handling availability. The notion of variable sublot is introduced explicitly by Potts & Baker [21]. For the 2-machine flow shop problem, they show that solving consistent sublot problem yields the optimal solution. They provide a counter example for 4-machine flow shop showing sub-optimality of consistent sublot solutions. They also derive several performance bounds identifying the sub-optimality of equal sublot solutions with respect to consistent sublot optimal solutions.

2.2.4 Three-Machine Problem

The three machine problem is first studied by Baker [4]. He derives explicit solution scheme for 2-sublot 3-machine flow shop problem. Trietsch & Baker [29] provide a three machine example where the consistent sublot solution is suboptimal. Baker & Jia [3] presents an experimental study to find the performance of the equal and consistent sublots solutions with respect to variable sublots solution.

Glass *et al.* [13] study lot streaming problem in three-machine flow shop, job shop and open shop to find the best consistent sublot solution. For the three machine flow shop problem they give a complete characterization of the optimal solution. They first analyze the critical path structure of the problem and identify two different patterns depending on the processing times of machines.

$$p_2^2 \leq p_1 p_3 \tag{2.3}$$

$$p_2^2 > p_1 p_3 \tag{2.4}$$

For the case (2.3), the optimal solution has the following form

$$L_k = \pi L_{k-1}$$

where

$$L_1 = \frac{1}{\sum_{l=0}^{s-1} \pi^l}$$

$$\pi = \frac{p_1 + p_2}{p_2 + p_3}$$

For the case (2.4), the sizes of the sublots depend on the crossover sublot l, that is, the sublot on which the critical path defining the makespan of the schedule jumps from the first machine to the second and third. They prove that in this case the optimal sublot sizes are in the form

$$L_k^{(l)} = \begin{cases} \pi_1^{l-k} L_l^{(l)} & \text{for } 1 \leq k \leq l \\ \pi_3^{k-l} L_l^{(l)} & \text{for } k \leq l \leq s \end{cases} \tag{2.5}$$

where

$$\pi_1 = \frac{p_1}{p_2}, \qquad \pi_3 = \frac{p_3}{p_2} \tag{2.6}$$

and

$$L_l^{(l)} = \frac{1}{\sum_{k=0}^{l-1} \pi_1^k + \sum_{k=0}^{s-l} \pi_3^k - 1} \tag{2.7}$$

Furthermore, they show that the sublot completion time is a convex function of l, and therefore, the value of l can be found in $O(\log\ s)$ time by bi-section search. The overall complexity of their algorithm is $O(s)$.

2.2.5 Special Cases of m-Machine Problems

The optimal sublot sizes given by the formula (2.2) are in the form of a geometric series. The resulting sublot fractions has an interesting characteristic. All sublots are critical, that is, there is no idle time in between completion of sublot k and the start of sublot $k+1$ (Figure 2.1).

The geometric sublot sizes can be shown to give optimal sublot sizes for a special case of m machine flow shop problem. First such result is due to Baker [4] for three machines two sublots problem. When $p_1\, p_3 \geq p_2^2$, sublot fractions can be computed using 2.2 and taking π as

$$\pi = \frac{p_2 + p_3}{p_1 + p_2}$$

This result is extended to the s sublots case in Glass *et al.* [13]. It can be shown that it can be generalized further to the m-machine single job problem:

Result 2.1 *Consider m machine streaming problem under consistency assumption with s transfer batches in between any consecutive stages. If*

$$p_1 \; p_m \geq (\sum_{i=2}^{m-1} p_i)^2 \tag{2.8}$$

then the optimal consistent sublot sizes are given as

$$L_k = \frac{\pi^{k-1}}{\sum_{l=0}^{s-1} \pi^l} \tag{2.9}$$

where

$$\pi = \frac{\sum_{i=2}^{m} p_i}{\sum_{i=1}^{m-1} p_i} \tag{2.10}$$

To be able to prove the result, we need to give the proof of following two results. In the proof of the Result 2.2 we make use of the network representation of the streaming problems, introduced by Baker & Pyke [5]. In this representation a node (i, k) corresponds k'th sublot on machine i. The arcs emanating from the node (i, k) have length $p_i \; L_k$. Assuming that L_k's are known in extent, the longest path in the network corresponds the makespan of the solution $L_k, k = 1 \ldots s$.

Result 2.2 *For any choice of sublot sizes, L_l, $l = 1 \ldots s$, the critical path defining the optimal makespan contains no more than one sublot on the machine i, $i = 2 \ldots m-1$.*

Proof: Assume that, for the particular sublot sizes L_l, $l = 1 \ldots s$, the critical path defining the makespan contains more than one sublot on some of middle machines (i, $i = 2 \ldots m-1$). Let k_i, $i = 1 \ldots m-1$ denote the index of the sublot through which the critical path jumps from machine i to machine $i+1$. In Figure 2.2, a particular configuration for the network representation of a 5-machine s-sublot problem is presented. Let CP^* denote the part of critical path from node $(1, k_1)$ to node (m, k_{m-1}). Then the CP^* is longer than both CP_1 and CP_2.

$$CP_1 < CP^*$$

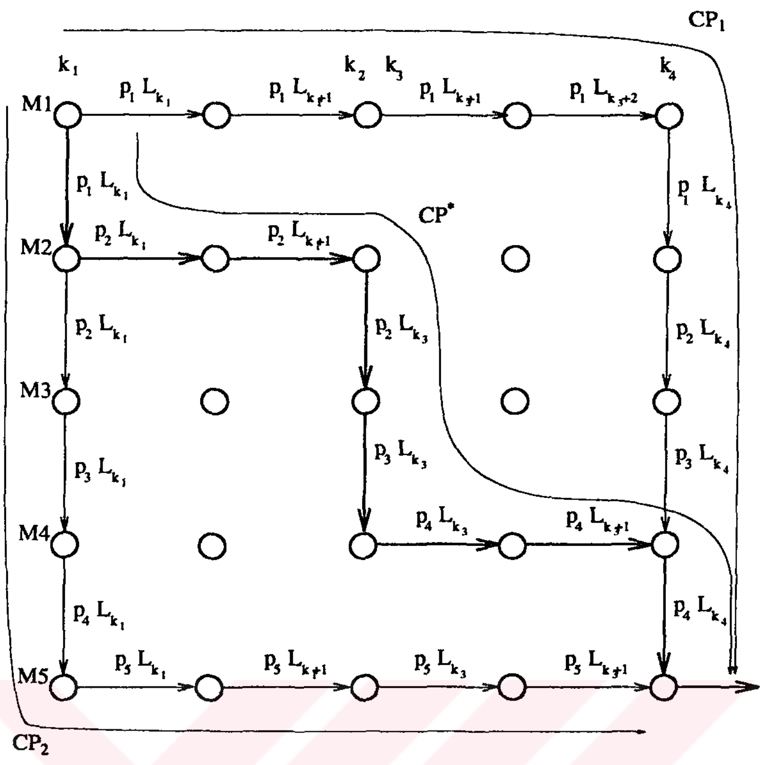

Figure 2.2: Network representation of the 5-machine problem

$$L_{k_{m-1}} \sum_{i=2}^{m-1} p_i + p_1 \sum_{l=k_1}^{k_{m-1}} L_l < p_1 L_{k_1} + \sum_{i=2}^{m-1} p_i \sum_{l=k_{i-1}}^{k_i} L_l$$

$$L_{k_{m-1}} \sum_{i=2}^{m-1} p_i + p_1 \sum_{l=k_1+1}^{k_{m-1}} L_l < \sum_{i=2}^{m-1} p_i \sum_{l=k_{i-1}}^{k_i} L_l$$

$$p_1 < \frac{\sum_{i=2}^{m-1} p_i \left(\sum_{l=k_{i-1}}^{k_i} L_l - L_{k_{m-1}} \right)}{\sum_{l=k_1+1}^{k_{m-1}} L_l} \tag{2.11}$$

$$CP_2 < CP^*$$

$$L_{k_1} \sum_{i=1}^{m-1} p_i + p_m \sum_{l=k_1}^{k_{m-1}-1} L_l < \sum_{i=2}^{m-1} p_i \sum_{l=k_{i-1}}^{k_i} L_l + p_1 L_{k_1}$$

$$L_{k_1} \sum_{i=2}^{m-1} p_i + p_m \sum_{l=k_1}^{k_{m-1}-1} L_l < \sum_{i=2}^{m-1} p_i \sum_{l=k_{i-1}}^{k_i} L_l$$

$$p_m < \frac{\sum_{i=2}^{m-1} p_i \left(\sum_{l=k_{i-1}}^{k_i} L_l - L_{k_1}\right)}{\sum_{l=k_1}^{k_{m-1}-1} L_l} \tag{2.12}$$

From (2.11) and (2.12) we obtain,

$$p_1 \; p_m < \frac{\left[\sum_{i=2}^{m-1} p_i \left(\sum_{l=k_{i-1}}^{k_i} L_l - L_{k_{m-1}}\right)\right] \; \left[\sum_{i=2}^{m-1} p_i \left(\sum_{l=k_{i-1}}^{k_i} L_l - L_{k_1}\right)\right]}{\left(\sum_{l=k_1}^{k_{m-1}-1} L_l\right) \; \left(\sum_{l=k_1+1}^{k_{m-1}} L_l\right)}$$

But clearly,

$$\left(\sum_{l=k_{i-1}}^{k_i} L_l - L_{k_{m-1}}\right) \leq \sum_{l=k_1+1}^{k_{m-1}} L_l$$

and

$$\left(\sum_{l=k_{i-1}}^{k_i} L_l - L_{k_1}\right) \leq \sum_{l=k_1}^{k_{m-1}-1} L_l$$

Therefore,

$$p_1 \; p_m < \left(\sum_{i=2}^{m-1} p_i\right)^2 \tag{2.13}$$

resulting in a contradiction. □

For a given sublot fractions, we call the sublot c critical if

$$c = \text{argmax}_{\{1 \leq k \leq s\}} \left\{ p_1 \sum_{l=1}^{k-1} L_l + L_k \sum_{i=1}^{m} p_i + p_m \sum_{l=k+1}^{s} L_l \right\}$$

Result 2.3 *If the processing times satisfy the inequality 2.8 then in the optimal solution, there cannot be a non-critical sublot.*

Proof: The proof we present here is a straightforward extension of the proof given by Potts & Baker [21] for 2-machine flow shop.

Let L_l, $l = 1 \ldots s$ and $F(L)$ be the optimal solution and optimal makespan for the problem. Moreover assume that there is an intermittent idleness after sublot L_k on the first or on the last machine. Consider following solution obtained by perturbing the optimal solution,

$$\begin{aligned} L_k^{'} &= L_k(1-\delta)+\delta \\ L_l^{'} &= L_l(1-\delta) \qquad\qquad l \neq k \end{aligned}$$

and define Δ as

$$\Delta = F(L) - \left[p_1 \sum_{l=1}^{k-1} L_l + L_k \sum_{i=1}^{m} p_i + p_m \sum_{l=k+1}^{s} L_l\right]$$

For the perturbed solution let $c^{'}$ be the index of critical sublot, that is

$$F(L^{'}) = p_1 \sum_{l=1}^{c^{'}-1} L_l^{'} + L_{c^{'}} \sum_{i=1}^{m} p_i + p_m \sum_{l=c^{'}+1}^{s} L_l^{'} \tag{2.14}$$

If $c^{'} < k$, we have

$$F(L^{'}) = p_1(1-\delta) \sum_{l=1}^{c^{'}-1} L_l + L_{c^{'}}(1-\delta) \sum_{i=1}^{m} p_m + p_m(1-\delta) \sum_{l=c^{'}+1}^{s} L_l + p_1\delta \tag{2.15}$$

If $c^{'} > k$, then

$$F(L^{'}) = p_1(1-\delta) \sum_{l=1}^{c^{'}-1} L_l + L_{c^{'}}(1-\delta) \sum_{i=1}^{m} p_m + p_m(1-\delta) \sum_{l=c^{'}+1}^{s} L_l + p_m\delta \tag{2.16}$$

From (2.15) & (2.16) we get

$$F(L^{'}) \leq p_1(1-\delta) \sum_{l=1}^{c^{'}-1} L_l + L_{c^{'}}(1-\delta) \sum_{i=1}^{m} p_m + p_m(1-\delta) \sum_{l=c^{'}+1}^{s} L_l + \delta \max\{p_1, p_m\}$$

$$F(L') \leq (1-\delta)F(L) + \delta \ \max\{p_1, p_m\}$$

But since $\max\{p_1, p_m\} < F(L)$,

$$F(L) > F(L')$$

For $c' = k$, from (2.14) we have,

$$F(L') = (1-\delta) \ \left[p_1 \sum_{l=1}^{k} L_l + L_k \sum_{i=1}^{m} p_i + p_m \sum_{l=k}^{s} L_l\right] + \delta \sum_{i=1}^{m} p_i$$

$$F(L') = (1-\delta)\,(F(L) - \Delta)) + \delta \sum_{i=1}^{m} p_i$$

$$F(L') = F(L) + \delta \left(\sum_{i=1}^{m} p_i - F(L) + \Delta\right) - \Delta$$

For the following choice of δ

$$\delta < \frac{\Delta}{\sum_{i=1}^{m} p_i - F(L) + \Delta}$$

again we get,

$$F(L) > F(L') \qquad \square$$

Proof [Result 2.1]: It follows from the Result 2.3 that for an optimal solution L_l, $l = 1 \dots s$, we have

$$F(L) = p_1 \sum_{l=1}^{c-1} L_l + L_c \sum_{i=1}^{m} p_i + p_m \sum_{l=c+1}^{s} L_l \qquad (2.17)$$

$$F(L) = p_1 \sum_{l=1}^{c} L_l + L_{c+1} \sum_{i=1}^{m} p_i + p_m \sum_{l=c+2}^{s} L_l \qquad (2.18)$$

From (2.17) & (2.18) we get,

$$p_1 L_c + (L_{c+1} - L_c) \sum_{i=1}^{m} p_i - p_m L_{c+1} = 0$$

$$L_{c+1} \sum_{i=1}^{m-1} p_i = L_c \sum_{i=2}^{m} p_i$$

$$\frac{L_{c+1}}{L_c} = \frac{\sum_{i=2}^{m} p_i}{\sum_{i=1}^{m-1} p_i} = \pi$$

Since $\sum_{l=1}^{s} L_l = 1$, we have

$$L_1 = \frac{1}{\sum_{l=0}^{s-1} \pi^l}$$

and

$$L_k = \frac{\pi^{k-1}}{\sum_{l=0}^{s-1} \pi^l} \qquad \square$$

2.3 Sublot Completion Time Criterion

The sublot and item completion time criteria result in quadratic programming models, hence they are expected to be harder than the job completion time problems, which are known to be solvable in polynomial time under the consistency assumption [4].

For the sublot completion time criterion, several quadratic programming formulations are given in Kropp & Smunt [16]. Their study emphasize the experimental findings, rather than specific analytical results. The analytical treatment of several problems under this criterion is presented in Topaloğlu *et al.* [27], Şen *et al.* [26] and Çetinkaya & Gupta [9]. The optimality proof of equal sublots solution when the first machine has the greatest processing time, is given in Şen *et al.* [26] and Çetinkaya & Gupta [9]. Şen *et al.* [26] reveal some interesting characteristics of two machine problem, and give an $O(s)$ algorithm to find the best consistent sublots, along with an example showing the sub-optimality of the consistent sublot solutions in a two-stage shop. They also give performance bounds for the equal sublot solutions.

Çetinkaya & Gupta [9] present an $O(m^2)$ algorithm for the two sublot problem under sublot completion time criterion. Their algorithm exploits the feasible machine concept proposed by Baker and Pyke [5] and the results from the ordered flow shops of the classical scheduling theory.

In the following section we give general results for sublot completion time criterion. In Section 2.3.3 we derive two $O(m^2)$ algorithms for m-machine two sublot problem. We present results pertaining 2-machine s-sublot problem in Section 2.3.4.

2.3.1 The Problem

We consider the following formulation of lot streaming problem with mean flow time criterion under consistency assumption,

$$
\begin{array}{lll}
\min & F(L) = \sum_{k=1}^{s} L_k \, C_{mk} & \\
\text{subject to} & \sum_{k=1}^{s} L_k = U & \\
& C_{11} - p_1 \, L_1 \geq 0 & \\
& C_{ik} - C_{i,k-1} - p_i \, L_k \geq 0, & i = 1, \dots, m, \quad k = 2, \dots, s \\
& C_{ik} - C_{i-1,k} - p_i \, L_k \geq 0, & i = 1, \dots, m, \quad k = 2, \dots, s \\
& C_{ik} \geq 0, & i = 1, \dots, m, \quad k = 2, \dots, s \\
& L_k \geq 0, & k = 1, \dots, s
\end{array}
$$

where C_{ik} denotes the completion time of sublot k on machine i, L_k denotes the size of sublot k, and U denotes the number of items in the given job. The objective function value, the total flow time denoted by $F(L)$, is the function of the sublot sizes. Without loss of generality, we let $U = 1$.

Since a sublot can be removed from the shop only after the whole sublot is completed on the last machine, in the objective function the completion time of sublot on the last machine is weighed with the proportion of lot in the corresponding sublot.

2.3.2 General Results

Following special case covers a considerable fraction of flow shop streaming problems.

Result 2.4 *In a single job lot streaming problem, if the following condition holds,*

$$p_1 = \max_{\{1 \leq i \leq m\}} p_i$$

then the sublots of equal size are optimal, i.e.,

$$L_k = \frac{1}{s} \qquad k = 1, \ldots, s$$

We first need the following result showing that there exists an optimal solution with nondecreasing sublot sizes[1].

Result 2.5 *If* $p_1 = \max_{1 \leq i \leq m}\{p_i\}$ *then an optimal solution exists in which,*

$$L_k \leq L_{k+1}, \qquad k = 1, \ldots, s. \tag{2.19}$$

Proof: Suppose the contrary, that is, there exists an optimal solution $L = [L_1, \ldots, L_s]$ such that for at least one k , $L_k > L_{k+1}$.

[1]The property given in equation (2.19) is proved for any choice of the processing times by Çetinkaya & Gupta [9].

Now we will give an algorithm that will construct a schedule satisfying Condition (2.19) and having the objective value not more than that of L. Let $\Pi_t = (\Pi_t(1), \Pi_t(2), \ldots, \Pi_t(s))$ and $\overline{\Pi} = (\overline{\Pi}(1), \overline{\Pi}(2), \ldots, \overline{\Pi}(s))$ denote the sublot sequence at t^{th} iteration of the algorithm and the optimal sublot sequence, respectively.

```
$\Pi_0 \leftarrow \overline{\Pi}$
for $t = 0$ to $s-1$ do
  begin
    $r \leftarrow \arg\max_{\{1 \le k \le s-t\}}\{L_{\Pi_t(k)}\}$
    $\Pi_{t+1} \leftarrow \Pi_t$
    for $k = r$ to $s-t$ do
      $\Pi_{t+1}(k) \leftarrow \Pi_t(k+1)$
    $\Pi_{t+1}(s-t) \leftarrow \Pi_t(r)$
  end
```

In the t^{th} iteration of first "*for*" loop, the minimum sublot among the first $(s-t)$ sublots in the sequence Π_t is removed from its place and inserted in $(s-t)^{th}$ place to form the sequence Π_{t+1}. The final schedule satisfies,

$$L_k \le L_{k+1}$$

To show that the resulting schedule has objective value not more than the optimal solution, consider the following two observations,

Observation 1 *In the t^{th} iteration of the algorithm, if the largest sublot among the first $(s-t)$ sublots, $\Pi_t(r)$, is removed from the schedule, then the minimum decrease in the total flow time Δ^- is,*

$$\Delta^- \ \ge\ p_1 L_{\Pi_t(r)} \sum_{l=r+1}^{s} L_{\Pi_t(l)}$$

Proof: Let C^- be the minimum decrease in the completion time of the sublots that follow the removed sublot. Clearly,

$$\Delta^- \geq C^- \sum_{l=r+1}^{s} L_{\Pi_t(l)}.$$

As illustrated in Figure 2.3, a lower bound on the C^- can be found as,

$$C^- \geq \min_{\{1\leq i\leq m\}} \left\{C_{i,\Pi_t(r)} - C_{i,\Pi_t(r-1)}\right\}.$$

As it is mentioned in Section 2.2.1, the lot streaming problem turns out to be an ordered flow shop problem, when the sublot sizes are given. In ordered flow shops, when the first machine has the largest processing time, the minimum makespan is achieved, if the jobs are in non-increasing order of processing times. This result is given by Smith *et al* [24]. In order to prove this result, they consider an optimal sequence with at least one job whose processing time is larger than the processing time of the immediately succeeding job. They show that, the new sequence formed by pairwise interchange of these two jobs does not have longer makespan. This implies, however, that the longest makespan is achieved when the jobs are in nondecreasing order of processing times. Using this fact, we get

$$C_{i,\Pi_t(r-1)} \leq p_1 \sum_{l=1}^{r-1} L_{\Pi_t(l)} + \max_{\{1\leq l\leq r-1\}}\{L_{\Pi_t(l)}\} \sum_{v=2}^{i} p_v.$$

The expression on the right is the maximum completion time that can be achieved on the i^{th} machine by sequencing the first $i-1$ sublots. The following is a lower bound on $C_{i,\Pi_t(r)}$,

$$C_{i,\Pi_t(r)} \geq p_1 \sum_{l=1}^{r} L_{\Pi_t(l)} + L_{\Pi_t(r)} \sum_{v=2}^{i} p_v.$$

Thus, we have

$$\begin{aligned} C^- &\geq \min_{\{1\leq i\leq m\}} \Big\{p_1 \textstyle\sum_{l=1}^{r} L_{\Pi_t(l)} + L_{\Pi_t(r)} \sum_{v=2}^{i} p_v \\ &\quad -(p_1 \textstyle\sum_{l=1}^{r-1} L_{\Pi_t(l)} + \max_{\{1\leq l\leq r-1\}}\{L_{\Pi_t(l)}\} \cdot \sum_{v=2}^{i} p_v)\Big\}, \end{aligned}$$

$$\geq \min_{\{1\leq i\leq m\}} \left\{ p_1 L_{\Pi_t(r)} + \left(L_{\Pi_t(r)} - \max_{\{1\leq l\leq r-1\}} \{L_{\Pi_t(l)}\} \right) \cdot \textstyle\sum_{v=2}^{i} p_v \right\}.$$

Since $L_{\Pi_t(r)} = \max_{\{1\leq l\leq r\}}\{L_{\Pi_t(l)}\} \geq \max_{\{1\leq l\leq r-1\}}\{L_{\Pi_t(l)}\}$ we have

$$C^- \geq \min_{\{1\leq i\leq m\}}\{p_1 L_{\Pi_t(r)}\},$$

$$\geq p_1 L_{\Pi_t(r)}.$$

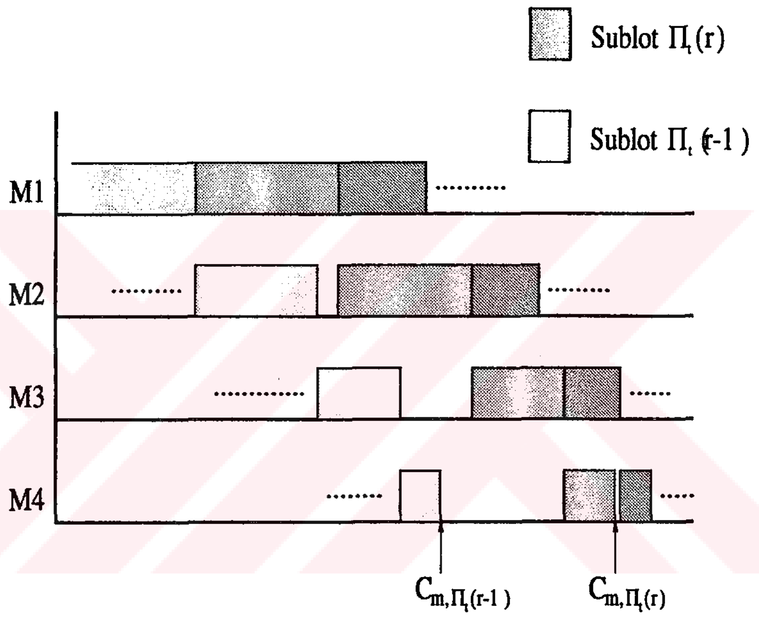

Figure 2.3: Gantt Chart of Π_t

Hence, the minimum decrease in the objective function is

$$\Delta^- \geq p_1 L_{\Pi_t(r)} \sum_{l=r+1}^{s} L_{\Pi_t(l)} \quad \square$$

Observation 2 *If the largest sublot among the first t sublots, $\Pi_t(r)$, is inserted $(s-t)^{th}$ place in the schedule (Figure 2.4), then the maximum increase in the total flow time Δ^+ is,*

$$\Delta^+ \leq p_1 L_{\Pi_t(r)} \sum_{l=r+1}^{s} L_{\Pi_t(l)}$$

Proof: Let C^+ be the increase in the completion time of removed sublot, when it is inserted in the $(s-t)^{th}$ place.

Observe that,

$$\Delta^+ = L_{\Pi_t(r)} C^+ + p_1 L_{\Pi_t(r)} \sum_{l=s-t+1}^{s} L_{\Pi_t(l)}$$

C^+ can be written as

$$\begin{aligned} C^+ &\leq C_{i,\Pi_{t+1}(s-t)} - C_{i,\Pi_t(r)} \\ &\leq p_1 \textstyle\sum_{l=1}^{s-t} L_{\Pi_t(l)} + L_{\Pi_t(r)} \sum_{v=2}^{m} p_v - (p_1 \sum_{l=1}^{r} L_{\Pi_t(l)} + L_{\Pi_t(r)} \sum_{v=2}^{m} p_v) \\ &\leq p_1 \textstyle\sum_{l=r+1}^{s-t} L_{\Pi_t(l)} \end{aligned}$$

Hence, the maximum increase is

$$\begin{aligned} \Delta^+ &\leq p_1 L_{\Pi_t(r)} \textstyle\sum_{l=r+1}^{s-t} L_{\Pi_t(l)} + p_1 L_{\Pi_t(r)} \sum_{l=s-t+1}^{s} L_{\Pi_t(l)} \\ &\leq p_1 L_{\Pi_t(r)} \textstyle\sum_{l=r+1}^{s} L_{\Pi_t(l)} \qquad \square \end{aligned}$$

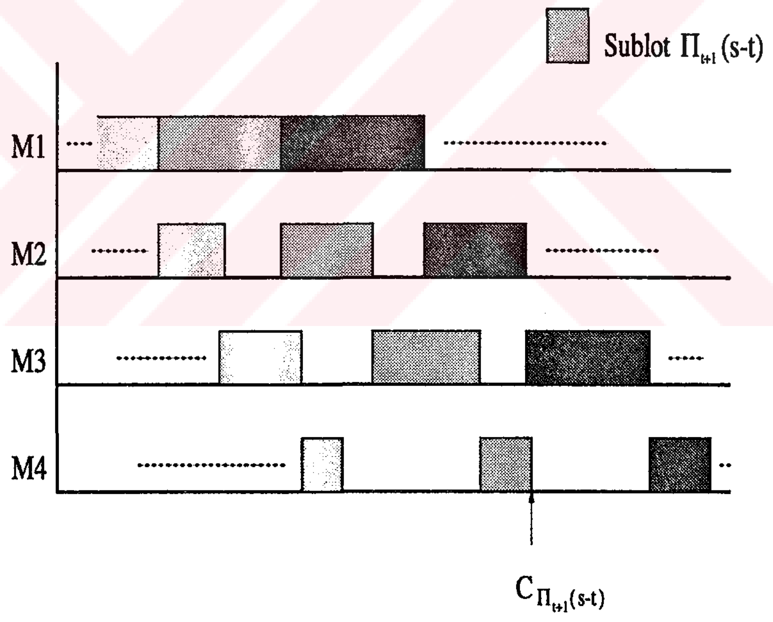

Figure 2.4: Gantt Chart of Π_{t+1}

Therefore, in the t^{th} step of proposed algorithm, the maximum overall increase in the mean flow time value is

$$\Delta^+ - \Delta^- \leq 0$$

Hence, the mean flow time of the sublot schedule constructed by the algorithm is not worse. Thus, in the optimal solution, $L_k \leq L_{k+1}$, $k = 1, \ldots, s$.

As shown in Figure 2.4, the completion time of the sublot k on the last machine is,

$$C_{mk} = p_1 \sum_{l=1}^{k-1} L_l + L_k \sum_{i=1}^{m} p_i \ .$$

With this property the following concise formulation with a convex objective function and fewer constraints can be obtained,

$$\begin{array}{lll} \min & \sum_{k=1}^{s} L_k C_{mk} & \\ \text{st} & C_{mk} - p_1 \sum_{l=1}^{k-1} L_l - L_k \sum_{i=1}^{m} p_i = 0 & k = 1, \ldots, s \\ & \sum_{k=1}^{s} L_k = 1 & \end{array}$$

or, equivalently,

$$\begin{array}{ll} \min & \sum_{k=1}^{s} L_k (p_1 \sum_{l=1}^{k-1} L_l + L_k \sum_{i=1}^{m} p_i) \\ \text{st} & \sum_{k=1}^{s} L_k = 1 \end{array}$$

Proof [Result 2.4]: The Lagrangian function of the problem is,

$$\mathcal{L}(L_1, \ldots., L_s, \delta) = \sum_{k=1}^{s} L_k (p_1 \sum_{l=1}^{k-1} L_l + L_k \sum_{i=1}^{m} p_i) + \delta(\sum_{k=1}^{s} L_k - 1),$$

then,

$$\frac{\partial \mathcal{L}}{\partial L_r} = p_1 \sum_{l=1}^{s} L_l + 2L_r \sum_{i=1}^{m} p_i - p_1 L_r + \delta = 0$$

and,

$$\frac{\partial \mathcal{L}}{\partial \delta} = \sum_{r=1}^{s} L_r - 1 = 0$$

Since,

$$\frac{\partial \mathcal{L}}{\partial L_r} - \frac{\partial \mathcal{L}}{\partial L_{r+1}} = 2(L_r - L_{r+1})\sum_{i=1}^{m} p_i - p_1 L_r + p_1 L_{r+1} = 0$$

or,

$$L_r(2\sum_{i=1}^{m} p_i - p_1) = L_{r+1}(2\sum_{i=1}^{m} p_i - p_1),$$

$$L_r = L_{r+1}$$

But $\sum_{k=1}^{s} L_k = 1$ implies that $L_r = 1/s$ is the candidate optimal solution. However, to prove that it is the desired solution, we have to show that the objective function is convex. The Hessian matrix of the objective function is

$$\begin{bmatrix} a & b & b & b & b & .. \\ b & a & b & b & b & .. \\ b & b & a & b & b & .. \\ b & b & b & a & b & .. \\ b & b & b & b & a & .. \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{bmatrix}$$

where

$$a = 2\sum_{i=1}^{m} p_i, \quad \text{and}$$
$$b = p_1$$

The positive definiteness of the Hessian matrix will imply the convexity of the objective function. In order to prove that a matrix is positive definite, it is enough to show that the diagonal elements of the U matrix in LU decomposition of Hessian matrix (or, the pivot elements without row exchanges) are all positive.

Consider any matrix with above structure with $a > b \geq 0$. After the first Gaussian elimination step, we get the first pivot entry as $(a > 0)$, with updated matrix

$$\begin{bmatrix} a & b & b & b & b & .. \\ 0 & \frac{a^2-b^2}{a} & \frac{a\cdot b-b^2}{a} & \frac{a\cdot b-b^2}{a} & \frac{a\cdot b-b^2}{a} & .. \\ 0 & \frac{a\cdot b-b^2}{a} & \frac{a^2-b^2}{a} & \frac{a\cdot b-b^2}{a} & \frac{a\cdot b-b^2}{a} & .. \\ 0 & \frac{a\cdot b-b^2}{a} & \frac{a\cdot b-b^2}{a} & \frac{a^2-b^2}{a} & \frac{a\cdot b-b^2}{a} & .. \\ 0 & \frac{a\cdot b-b^2}{a} & \frac{a\cdot b-b^2}{a} & \frac{a\cdot b-b^2}{a} & \frac{a^2-b^2}{a} & .. \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{bmatrix}$$

Since

$$\frac{a^2-b^2}{a} > \frac{a\cdot b-b^2}{a} > 0$$

the sub-matrix starting from second row and second column has the same structure as the original one. Hence, its first pivot element will be positive and the resulting matrix will have the common structure. The proof follows inductively. □

2.3.3 Two-Sublot Problem

Clearly, more the number of sublots in the shop, better the measure of performance is. But, the number of sublots that can be used for transfers is dictated by the material handling equipment availability in the shop, which is rarely unlimited. Analyzing the two sublot problem can be justified because of two reasons. First, the improvement in the objective function from the use of an additional sublot is expected to decrease as the number of sublots increases. Potts & Baker [21] shows that for the two machine flow shop problem under makespan criterion, the largest improvement is obtained when number of sublots is increased from one to two.

Second, since no practical algorithm is available yet for the m-stage s-sublot problem for sublot and item completion time criteria, the two sublot problem can be a building block in the design of heuristics to attack s-sublot problems.

The Model

The formulation of two sublot problem problem under sublot completion time criterion can be given as follows:

$$\begin{array}{lll}
\min\ F(L) = & [C_{m1}L_1 \ + \ C_{m2}L_2] & \\
\text{subject to} & L_1 + L_2 = U & \\
& C_{11} - p_1 L_1 \geq 0 & \\
& C_{i2} - C_{i1} - p_i L_2 \geq 0 & i = 1, \ldots, m \\
& C_{i1} - C_{i-1,1} - p_i L_1 \geq 0 & i = 2, \ldots, m\ ,\ k = 1, 2 \\
& C_{i2} - C_{i-1,2} - p_i L_2 \geq 0 & i = 2, \ldots, m\ ,\ k = 1, 2 \\
& L_1, L_2, C_{ik} \geq 0 &
\end{array}$$

The first constraint forces the entire job to be processed in its entirely. The second constraint defines the completion time of the first sublot on the first machine. The third constraint prevents the second sublot to be processed on a machine, before the first sublot is completed on the same machine. The last two constraints prevent the overlapping of operations of a sublot being processed on different machines. Without loss of generality, assume that the given job contains a single item ($U = 1$).

Note that (see Figure 2.5),

$$C_{m1} = L_1 \textstyle\sum_{l=1}^{m} p_l,$$
$$C_{m2} = \max_{\{1 \leq i \leq m\}} \{L_1 \textstyle\sum_{l=1}^{i} p_l + L_2 \sum_{l=i}^{m} p_l\},$$

and,

$$L_2 = 1 - L_1.$$

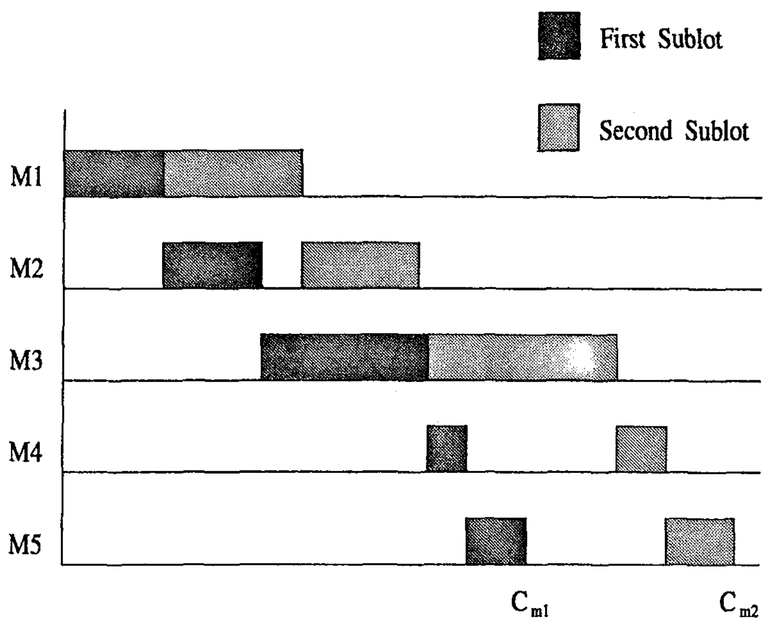

Figure 2.5: 5-machine shop with two consistent sublots

Thus, the objective function can be equivalently written as,

$$\min \quad F(L_1) = [L_1^2 \textstyle\sum_{l=1}^{m} p_l + \; (1 - L_1) \; \max_{1\leq i\leq m}\{L_1 \sum_{l=1}^{i} p_l + (1 - L_1) \; \sum_{l=i}^{m} p_l\}],$$

rearranging the terms,

$$\min \quad F(L_1) = [L_1^2 \textstyle\sum_{l=1}^{m} p_l + \max_{1\leq i\leq m}\{\sum_{l=1}^{i} p_l + L_1(\sum_{l=1}^{i} p_l - 2\sum_{l=i}^{m} p_l) + L_1^2(\sum_{l=i}^{m} p_l - \sum_{l=1}^{i} p_l)\}],$$

or, using the fact that maximum is the smallest upper bound,

$$\min F(L_1) \tag{2.20}$$

subject to

$$[(2\textstyle\sum_{l=i}^{m} p_l) - p_i]\, L_1^2 + \left[\sum_{l=1}^{i} p_l - 2\sum_{l=i}^{m} p_l\right] L_1 + \sum_{l=i}^{m} p_l - F(L_1) \leq 0,$$

$$i = 1, \ldots, m.$$

With these manipulations the problem reduces to a minimization problem containing only two variables, namely L_1 which is the fraction of the first sublot, and $F(L_1)$. There are m constraints each of which is a quadratic function of a single variable L_1.

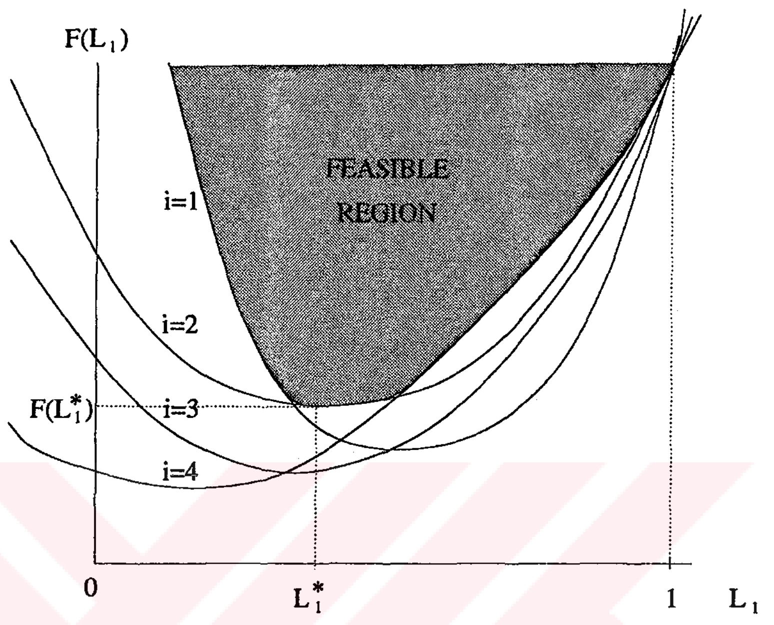

Figure 2.6: Feasible region of two-sublot the problem

The Analysis of the Model

Let a_i, b_i, c_i, denote the coefficients of L_1^2, L_1 and the constant term, respectively, in the Constraint i of (2.20). Since $a_i \geq 0$, $i = 1, \ldots, m$, $a_iL_1^2 + b_iL_1 + c_i$ is a convex function of L_1. The area defined by each constraint is the epigraph of the convex function, which is a convex set. Since the intersection of the convex sets is convex, the feasible region of the problem is convex. The objective function is linear, therefore the minimum will occur at the boundary of the feasible region. Furthermore, there are finite number of points that are candidates for optimum. This finite set of candidate points comprise $2 \cdot \binom{m}{2}$ intersection points of the constraints taken as equations and the minimum point of each individual equation, giving a total of m^2 candidate points for the optimum.

This search space can be reduced, by exploiting the following characteristics of the family of quadratic equations defining the feasible region.

Observation 3 *We first consider the intersection points of the feasible region. Consider two parabolas indexed as r and t. Without loss of generality, assume that $r > t$. The intersection points of these two parabolas are the roots of the quadratic equation*

$$a_{tr}L_1^2 + b_{tr}L_1 + c_{tr} = 0$$

where

$$\begin{aligned}
a_{tr} &= \left[(2\sum_{l=t}^{m} p_l) - p_t\right] - \left[(2\sum_{l=r}^{m} p_l) - p_r\right] \\
&= 2\sum_{l=t+1}^{r-1} p_l + p_r + p_t \\
b_{tr} &= (\sum_{l=1}^{t} p_l - 2\sum_{l=t}^{m} p_l) - (\sum_{l=1}^{r} p_l - 2\sum_{l=r}^{m} p_l) \\
&= -\sum_{l=t+1}^{r} p_l - 2\sum_{l=t}^{r-1} p_l \\
&= -3\sum_{l=t+1}^{r-1} p_l - p_r - 2p_t \\
c_{tr} &= (\sum_{l=t}^{m} p_l) - (\sum_{l=r}^{m} p_l) = \sum_{l=t+1}^{r-1} p_l + p_t
\end{aligned}$$

let $y \triangleq \sum_{l=t+1}^{r-1} p_l$ *then*

$$\begin{aligned}
a_{tr} &= 2y + p_r + p_t \\
b_{tr} &= -3y - p_r - 2p_t \\
c_{tr} &= y + p_t
\end{aligned}$$

The determinant of above quadratic equation is

$$\Delta_{tr} = b_{tr}^2 - 4a_{tr}c_{tr}$$

$$\begin{aligned}
&= (-3y - p_r - 2p_t)^2 - 4 \cdot (2y + p_r + p_t) \cdot (y + p_t) \\
&= y^2 + 2yp_r + p_r^2 \\
&= (y + p_r)^2
\end{aligned}$$

Then the roots of the equation are

$$\begin{aligned}
L_{1,1}^{tr} &= \frac{-b_{tr} - \sqrt{\Delta_{tr}}}{2a_{tr}} \\
&= \frac{3y + p_r + 2p_t - y - p_r}{2(2y + p_r + p_t)} \\
&= \frac{2y + 2p_t}{2(2y + p_r + p_t)} \\
&= \frac{y + p_t}{2y + p_r + p_t} \\
&= \frac{\sum_{l=t}^{r-1} p_l}{p_r + p_t + 2\sum_{l=t+1}^{r-1} p_l} \\
&= \frac{c_t - c_r}{a_t - a_r}
\end{aligned}$$

$$\begin{aligned}
L_{1,2}^{tr} &= \frac{-b_{tr} + \sqrt{\Delta_{tr}}}{2a_{tr}} \\
&= \frac{3y + p_r + 2p_t + y + p_r}{2(2y + p_r + p_t)} \\
&= \frac{4y + 2p_r + 2p_t}{4y + 2p_r + 2p_t} \\
&= 1 \qquad \square
\end{aligned}$$

Observation 4 *The parabola i intersects $F(L_1)$ axis at $\sum_{l=i}^{m} p_l$, implying that parabola $i+1$ intersects the $F(L_1)$ axis at a lower point than the parabola i.*

Observation 5 *The minimum point of parabola i occurs at*

$$L_1^{i*} = -\frac{b_i}{2a_i} = \frac{(2\sum_{l=i}^{m} p_l - \sum_{l=1}^{i} p_l)}{2(p_i + 2\sum_{l=i+1}^{m} p_l)}$$

$$L_1^{i*} = \frac{1}{2} - \frac{\sum_{l=1}^{i-1} p_l}{2(p_i + 2\sum_{l=i+1}^{m} p_l)}$$

which implies that as i increases the minimum values of parabolas lift to the left on the L_1-axis and all the minimal sublot fractions are less than 1/2.

Observation 6 *Since the minimum points of parabolas occur when $L_1 \leq 1/2$, all the parabolas have positive derivatives in the interval $L_1 \in (\frac{1}{2}, 1]$. Hence the size of the first sublot minimizing the sublot completion time criterion cannot be larger than $1/2$.*

Observation 7 *$a_i = (-p_i + 2\sum_{l=i}^{m} p_l) = (p_i + 2\sum_{l=i+1}^{m} p_l)$ is decreasing as i increases. Therefore the parabola $i+1$ is flatter than the parabola i.*

Observation 8 *The slope of the curve i at $L_1 = 1$,*

$$\begin{aligned} 2a_i + b_i &= 4(\textstyle\sum_{l=i}^{m} p_l) - 2p_i + \sum_{l=1}^{i} p_l - 2\sum_{l=i}^{m} p_l \\ &= 2\textstyle\sum_{l=i+1}^{m} p_l + \sum_{l=1}^{i} p_l \\ &= \textstyle\sum_{l=1}^{m} p_l + \sum_{i+1}^{m} p_l \end{aligned}$$

decreases as i increases.

Obviously, the candidate point $L_1 = 1$, which represents the case where no streaming is allowed, cannot be the optimal solution. Therefore, from the Observation 3, we conclude that, when the intersection points of two parabolas are considered as candidate points for the optimality, it suffices to check just one of the roots, namely to the nontrivial one,

$$\frac{c_t - c_r}{a_t - a_r}$$

This result reduces the number of candidate points resulting from the intersections of equations defined by the constraints by half.

The Algorithms

In this section we present two algorithms both having running time $O(m^2)$. First algorithm searches the optimum solution by moving from one candidate point to the adjacent one. Second one finds the optimal solution by performing *bi-section* search based procedure on the candidate optimal points. An experimental comparison of the algorithms is also presented.

Algorithm I

```
$t \leftarrow 1$
$CL \leftarrow \{2,,\ldots,,m\}$
$optimal \leftarrow false$
while not $optimal$ do
    begin
      $L_1 \leftarrow \frac{-b_t}{2a_t}$
      $r \leftarrow argmin_{\{i \in CL\}}\{\frac{c_t - c_i}{a_t - a_i}\}$
      if $\frac{c_t - c_r}{a_t - a_r} < L_1$ then
        begin
          $L_1 \leftarrow \frac{c_t - c_r}{a_t - a_r}$
          $CL \leftarrow CL - \{t\}$
          $t \leftarrow r$
          if $(2a_r L_1 + b_r) \geq 0$ then
            $optimal \leftarrow true$
          end
      else
        $optimal \leftarrow true$
    end.
```

The first constraint $(a_1L_1^2+b_1L_1+c_1)$ is taken as the initial parabola. Then, the algorithm visits the candidate optimal solutions of the feasible region. It checks the optimality of given solution by testing the condition

$$\frac{dF_r(L_1)}{dL_1} = (2a_rL_1 + b_r) \geq 0$$

where $F_r(L_1)$ represents the equation form of the constraint r. If this condition holds, then the intersection point is the first point that the objective function starts to increase. When the condition is satisfied with equality, then the given point is the minimal point of the parabola r. Hence its feasibility implies the optimality.

The next candidate point is determined, by finding the greatest L_1 value at which the current parabola is feasible. Apparently, this point is the smallest L_1 value at which the current parabola intersects with some other parabola. If more than one parabola intersect at the same point then the one with the smallest index is chosen as the search curve for the next iteration.

To prove the finite convergence of algorithm, it is enough to show that no candidate intersection point is visited more than once.

Result 2.6 *The algorithm visits a candidate optimal point at most once.*

Proof: In each iteration the algorithm chooses the parabola satisfying the condition,

$$r \leftarrow argmin_{\{i \in CL\}}\{\frac{c_t - c_i}{a_t - a_i}\}$$

Since the current parabola is removed from the candidate optimal list, (CL), and the next curve for candidate point search is chosen from this list, at the subsequent iterations the same point cannot be chosen. Hence, there cannot be any cycling taking place. The number of candidate optimal points is finite. □

Result 2.7 *The above algorithm converges to an optimal solution.*

Proof: The algorithm starts with the first constraint. As it can be easily inferred from the Observation 4, the first constraint is the only binding constraint in the interval $L_1 \in [-\infty, \epsilon]$, where $\epsilon \in [0, 1]$. The rule

$$r \leftarrow argmin_{\{i \in CL\}}\{\frac{c_t - c_i}{a_t - a_i}\}$$

is used find the index of the curve defining the next candidate point with the current parabola.

If the current iteration is not a degenerate one, obviously the next point is correctly determined, since beyond the first intersection point, the intersecting curve starts to be binding. Otherwise, the degenerate steps are repeated until the parabola with the largest index is chosen as the next curve to move on. Clearly, at the degenerate candidate point, the parabola with the largest

$$\frac{dF_i(L_1)}{dL_1}$$

value is the next binding curve. Hence, to complete the proof of the argument what we need to show is that the curve with the largest index has this property.

$$\begin{aligned}\tfrac{dF_r(L_1)}{dL_1} &= 2a_r L_1 + b_r \\ &= 2[(2\textstyle\sum_{l=r}^{m} p_l) - p_r]L_1 + \sum_{l=1}^{r} p_l - 2\sum_{l=r}^{m} p_l \\ &= \textstyle\sum_{l=1}^{r} p_l + 2(2L_1 - 1)\sum_{l=r}^{m} p_l - 2L_1 p_r\end{aligned}$$

hence,

$$\begin{aligned}\tfrac{dF_{r+1}(L_1)}{dL_1} - \tfrac{dF_r(L_1)}{dL_1} &= p_{r+1} - 2(2L_1 - 1)p_r - 2L_1(p_{r+1} - p_r) \\ &= (1 - 2L_1)p_{r+1} + (2 - 2L_1)p_r \\ &> 0.\end{aligned}$$

The last equality follows from the Observation 6, stating that in the optimal solution, the first sublot size (L_1) is less than or equal to one half. □

It is not difficult to show that, the coefficients, a_i, b_i, c_i, can be computed in linear time from the processing times. The algorithm performs $O(m)$ operations to find the next intersection point in the bottleneck operation

$$r \leftarrow argmin_{\{i \in CL\}}\{\frac{c_t - c_i}{a_t - a_i}\}$$

Hence, to be able to show that the complexity of algorithm is not more than $O(m^2)$, we need to prove that the number of candidate points resulting from the intersection of equations is in the order of m.

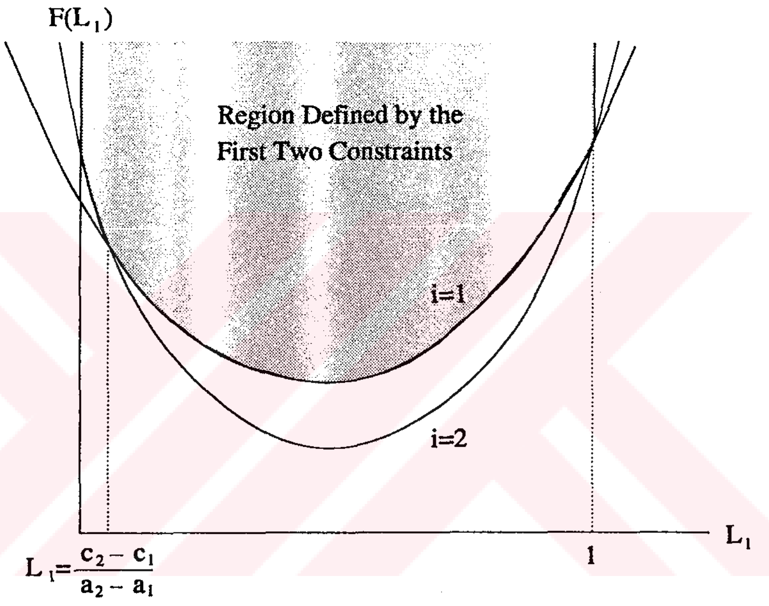
F(L₁)
Region Defined by the First Two Constraints
i=1
i=2
L₁
$L_1 = \frac{c_2 - c_1}{a_2 - a_1}$
1

Figure 2.7: Configuration when $n = 2$

Result 2.8 *Under sublot completion time assumption, the feasible region of the problem contains at most m intersection points.*

Proof: We prove this argument by induction on number of constraints. Consider a problem containing m machines,

$n = 2$ If only the region defined by the first two parabolas is considered, there can be at most two intersection points as illustrated in Figure 2.7.

$n = k$ Assume that the number of feasible intersection points is less than or equal to k, when there are k parabolas.

$n = k + 1$ Consider the new parabola added to the region defined by the first k parabolas. Let t be defined as,

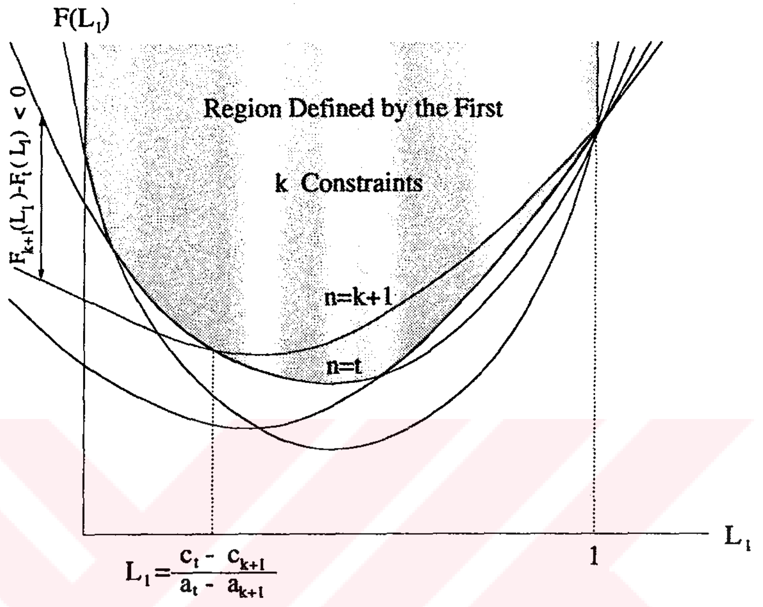

Figure 2.8: Configuration when $n = k + 1$

$$t \leftarrow \text{argmax}_{\{1 \leq i \leq k\}} \{ \frac{c_t - c_{k+1}}{a_t - a_{k+1}} \}$$

Namely t is the parabola that intersects with the parabola $k + 1$ at the largest L_1 value in the region $(0, 1)$. If the region defined by the first $k+1$ constraints are considered, it is not difficult to see from the Observation 8 that this intersection point is feasible, since it shows that in the region

$$L_1 \in [\frac{c_t - c_{k+1}}{a_t - a_{k+1}}, 1] \text{ ,}$$

the curve $k + 1$ is binding as shown in Figure 2.8. Let $F_i(L_1)$, denote the value of i'th constraint at L_1. Observe that

$$F_t(L_1) > F_{k+1}(L_1)$$

when

$$L_1 < \frac{c_t - c_{k+1}}{a_t - a_{k+1}}$$

since two distinct parabolas can intersect at most at two different points, the curves t and r cannot intersect when $L_1 < \frac{c_t - c_{k+1}}{a_t - a_{k+1}}$. Clearly the feasible region defined by the first k constraints is contained in the region $\{L_1 : F_t(L_1) \leq F(L_1)\}$. Hence, there cannot be any other feasible intersection point resulting from adding the $(k+1)$'th parabola (Figure 2.8). Therefore the number of intersection points is at most $k+1$, showing that the feasible region contains at most m intersection points. □

Algorithm II

The algorithm we present in this section is bi-section search based, and has the same worst case complexity bound as the first algorithm. Let $F_{k+1}(L_1)$ be defined as before and L_1^l, L_1^r be the right and left margins containing the optimal solution, respectively.

In each iteration the algorithm computes improving upper and lower bounds $[L_1^l, L_1^r]$ on the optimal solution. The statement,

$$d \leftarrow \text{argmax}_{\{1 \leq i \leq m\}}\{F_i(L_1^m)\}$$

finds the parabola that is binding at point $L_1^m = \frac{L_1^l + L_1^r}{2}$. If there are two parabolas with the same property, then the tie is broken by choosing the one with the largest index, since it is the binding curve on the relevant side of L_1^m. If the dominant parabola is increasing at point L_1^m, the nearest intersection point on the left is set as the new right limit as shown in Figure 2.9 . If the dominant parabola is decreasing at point L_1^m, the nearest point on the right is set as the new left limit. The algorithm terminates when right and left limits coincide, or the minimum point of the parabola d, the one that is binding at the point $L_1^m = \frac{L_1^l + L_1^r}{2}$, is feasible.

```
$L_1^l \leftarrow 0$
$L_1^r \leftarrow 1$
$optimal \leftarrow false$
while not $optimal$ do
  begin
    $L_1^m \leftarrow \frac{L_1^l + L_1^r}{2}$
    $d \leftarrow \text{argmax}_{\{1 \leq i \leq m\}}\{F_i(L_1^m)\}$
    if $2a_d L_1^m + b_d > 0$ then
      begin
        $L_1 \leftarrow \max_{\{i:\ i \neq d,\ \frac{c_d - c_i}{a_d - a_i} < L_1^m\}}\{\frac{c_d - c_i}{a_d - a_i}\}$
        if $(L_1 < -\frac{b_d}{2a_d})$ then
          begin
            $optimal \leftarrow true$
            $L_1^* \leftarrow -\frac{b_d}{2a_d}$
          end
        else $L_1^r \leftarrow L_1$
      end
    if $2a_d L_1^m + b_d < 0$ then
      begin
        $L_1 \leftarrow \max_{\{i:\ i \neq d,\ \frac{c_d - c_i}{a_d - a_i} > L_1^m\}}\{\frac{c_d - c_i}{a_d - a_i}\}$
        if $(L_1 > -\frac{b_d}{2a_d})$ then
          begin
            $optimal \leftarrow true$
            $L_1^* \leftarrow -\frac{b_d}{2a_d}$
          end
        else $L_1^l \leftarrow L_1$
      end
    if $L_1^l = L_1^r$ then
      begin
        $optimal \leftarrow true$
        $L_1^* \leftarrow L_1^r$
      end
  end.
```

To prove that the correctness of the algorithm, we need following two results.

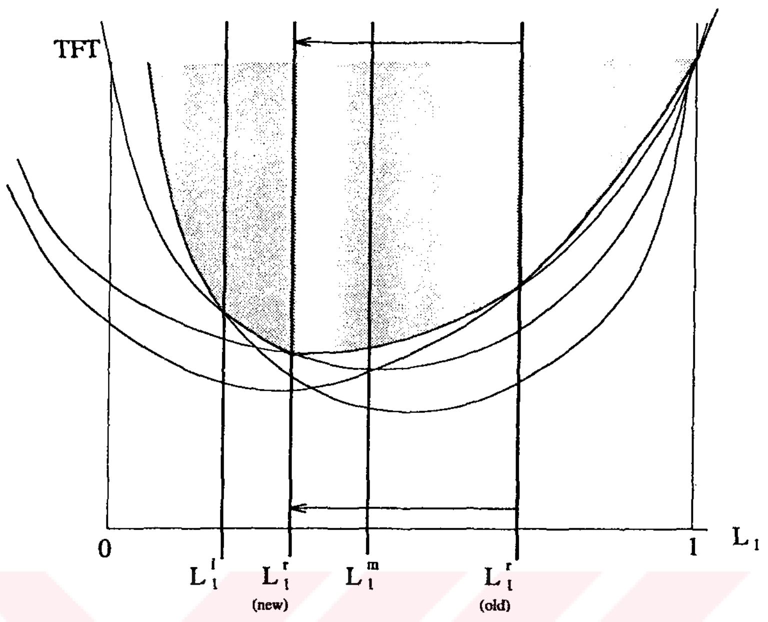

Figure 2.9: Updating L_1^r

Result 2.9 *The intervals $[L_l, L_r]$ computed in each iteration contains the optimal solution, L_1^*.*

Proof: Clearly, the initial interval $[L_l, L_r] = [0, 1]$ contains the optimal solution. In each iteration, one of the bounds is tightened according to the slope of the parabola d at point L_m. If the slope is positive [negative] then right [left] limit is decreased [increased] to the closest intersection point on the left [right]. If next intersection point is less [greater] than the point where the minimum value of the current parabola occurs, then algorithm terminates with the optimal solution as the minimum point of current parabola. Otherwise, the intersection point is updated as the new right (left) limit. Clearly, with these modifications, the optimal solution is retained within the limits. □

Result 2.10 *The algorithm terminates in at most m iterations.*

Proof: At each iteration, the algorithm finds either the optimal solution or discards at least one intersection point by updating one of the limits. Since the same intersection point cannot be repeated, and the feasible region contains at most m intersection points as shown in the Result 2.8, the algorithm terminates after investigating at most m intersection points. □

Result 2.11 *The Algorithm II is of $O(m^2)$ complexity.*

Proof: The coefficients of the parabolas can be computed in linear time. The bottleneck operations of the algorithm are the "*max*" and "*argmax*" statements, requiring $O(m)$ operations. Since there are only m intersection points, the while loop can iterate at most m times. Hence, the algorithm performs $O(m^2)$ operations in the worst case. □

The problems of streaming two sublots in an m-machine flow shop under makespan (job completion time) and sublot completion time criteria poses the similar characteristics. Under makespan criterion the feasible region is defined by linear equations rather than quadratic equations (Figure 2.10). The optimal solution is found by searching the upper envelope defined by m linear equations. Clearly, the minimum makespan achieved at one of the intersection points.

A slight modification of our algorithms can be used to solve makespan problem in $O(m^2)$ time. Baker & Pyke [5] make use of a heuristic argument that the line corresponding to the machine with the largest processing time is usually one of the binding curves at the optimal intersection point. Therefore, they find a near optimum solution. The other line is searched from the remaining lines having slope oppositely signed from the slope of the line corresponding to machine with the largest processing time. Let t be the index of the machine with the largest processing time. If the lines are in the form,

$$a_i L_1 + b_i, \quad i = 1, \ldots, m$$

then the index of the other binding line is,

$$r = \text{argmax}_{\{i: i \neq t,\ 1 \leq i \leq m,\ a_t \cdot a_i \leq 0\}} \left\{ a_t \left(\frac{b_t - b_i}{a_i - a_t} \right) + b_t \right\}$$

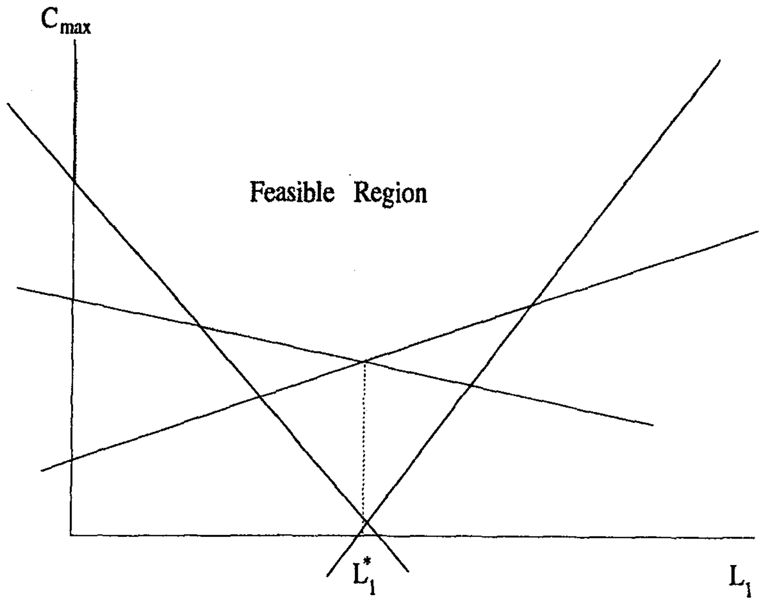

Figure 2.10: Feasible region of the makespan criterion

Experimental Study of Algorithms

As we have shown in the previous sections, the proposed algorithms have the same worst case complexity bounds. To compare their practical running times we carried out an experimental study. We considered eleven sets of problems consisting 1 to 255 machines. For each set, we generated 50,000 problems whose processing times were chosen randomly from a uniform distribution in the range $[1, 100]$. Each problem is solved by both algorithms and statistics are gathered separately. The results are listed in the following table.

The first column of the table lists the number of machines in the problem set. The second and third columns tabulates the maximum number of iterations performed to solve the 50,000 problems by Algorithms I and II, respectively. The fourth (fifth) column lists the maximum difference between the number of iterations performed by the algorithms, among the problems that the first (second) algorithm performed better. Finally, the last two columns lists average number of iterations performed by Algorithms I and II, respectively.

m	Max it I	Max it II	Max dif I	Max dif II	Avg it I	Avg it II
1	1	1	0	0	1	1
2	2	1	0	1	1.17	1
3	3	3	1	2	1.27	1.05
4	3	3	1	2	1.31	1.10
5	2	4	1	2	1.38	1.16
10	4	4	2	3	1.53	1.39
20	4	5	2	3	1.67	1.69
40	5	6	2	3	1.83	1.99
80	6	6	2	5	1.98	2.21
160	7	6	3	5	2.15	2.42
255	7	7	2	6	2.32	2.59

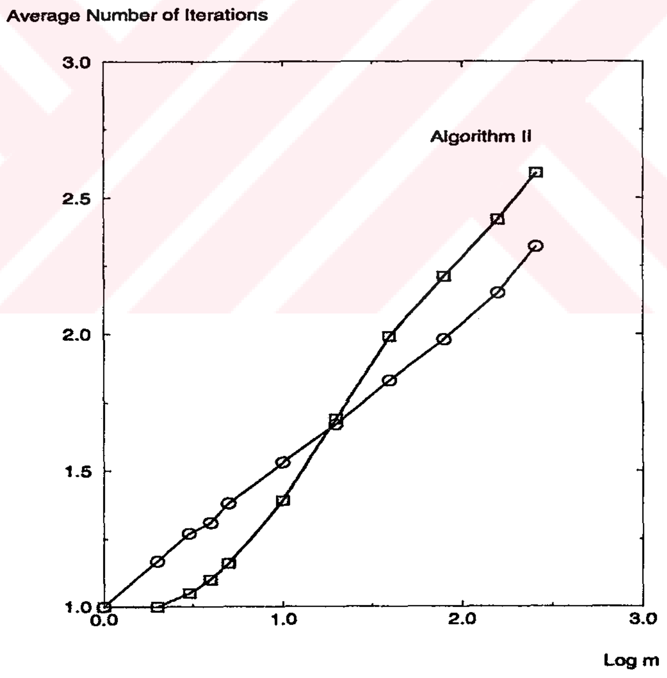

Figure 2.11: Average number of iterations performed by Algorithms I and II

Interestingly, the average of the algorithms are far better than their worst case bounds. Even for 255 machine problem set, all the problems are solved

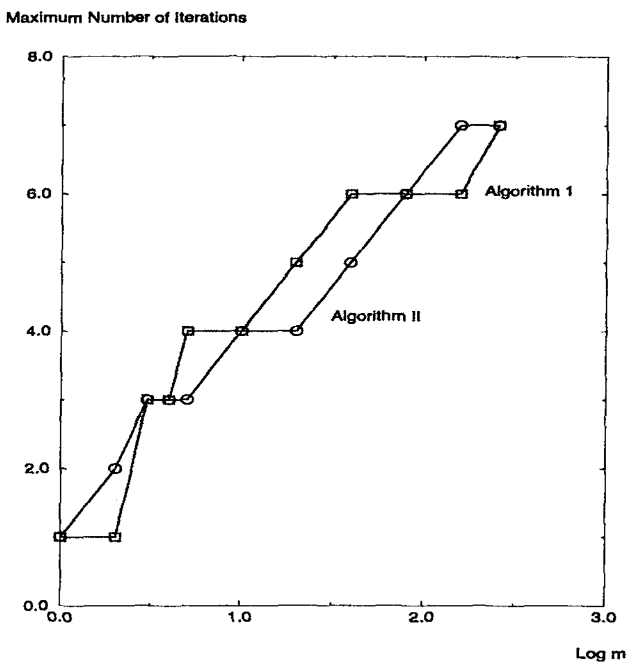

Figure 2.12: Maximum number of iterations performed by Algorithms I and II

within 7 iterations by both algorithms. On the average, 2.5 iterations of Algorithms I and II were required to find the optimum solutions for 255 machine problem set.

As shown in the first graph, up to 15 machines the second algorithm performs consistently better than the first one as seen in Figure 2.11. However, when there are more than 15 machines, the first algorithm dominates the second one on the average number of iterations performed. The maximum number of iterations needed to solve each set is plotted in the Figure 2.12. As the fifth and the sixth columns indicate, the variance of the second algorithm is smaller. There are problems solved in a single iteration by Algorithm II, which were solved in seven iterations by Algorithm I.

2.3.4 Two-Machine Problem

Although the two stage machining shops rarely exists in practice, they have received considerable attention because of their simplicity compared to the problems having three or more stages. The two machine problem may give insights about the characterization of the optimal solution of the m-machine problem. Furthermore heuristic procedures can be developed based on these characteristic to obtain good solutions for the general problem.

Sub-optimality of Consistent Sublots

In this study, we consider streaming problems under consistency and infinite divisibility assumptions. As stated before the consistency assumption may cause sub-optimality. However, for the makespan criterion (job completion time criterion), Potts & Baker [21] have shown that in a two machine flow shop, the consistent sublots yield the optimal solution. But this is not the case for the sublot completion time criterion. To show the sub-optimality of consistent sublots, consider the problem of streaming a job in a two machine flow shop with processing times $p_1 = 1$, $p_2 = 2$. Assume that only two sublots can be used on either machines (Figure 2.13). The best consistent sublot solution is,

$$L_1 = \frac{1}{3},\ L_2 = \frac{2}{3},\ F(L) = \frac{34}{18}$$

whereas the following variable sublots solution has better objective value,

$$L_{11} = \frac{1}{3},\ L_{12} = \frac{2}{3},\ L_{21} = \frac{1}{2},\ L_{22} = \frac{1}{2},\ F(L) = \frac{33}{18}$$

To justify the above argument intuitively, assume that we are given the sublot sizes on the first machine. Also assume that these sublots allow a continuous production on the second machine. Since one cannot vary the sublot sizes on the first machine the best strategy can be found by considering the second machine as a single machine. Therefore, the best one can do is to complete the job using equal transfer batches, as shown in Section 2.3.2.

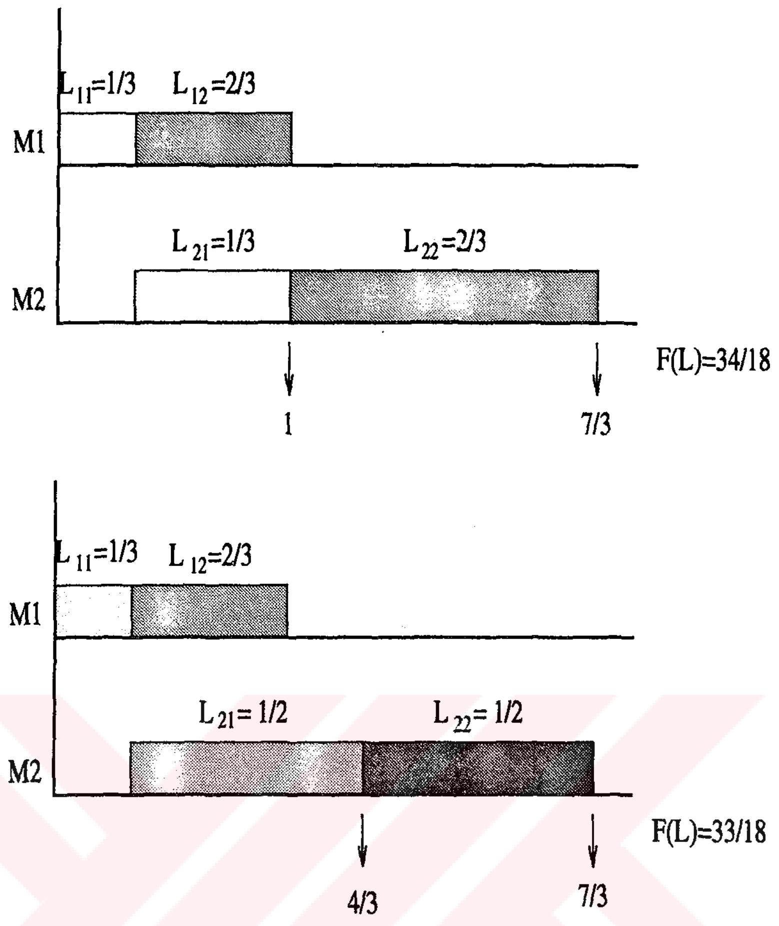

Figure 2.13: Optimal consistent sublot and variable sublot solutions

The Results

In two machine problem under sublot completion time criterion we consider two cases,

$$p_1 \geq p_2$$

$$p_1 \leq p_2$$

The solution for the first case is actually derived in Section 2.3.2 as

$$L_k = \frac{1}{s}, \qquad k = 1, \ldots, s$$

However the optimization problem in the second case is quite difficult as compared to the first one, because of the complicated nature of the constraints. The optimal solution is found in Şen *et al.*, [26] by observing the fact that in the optimal solution there cannot be an idle time in between the processing of any two consecutive sublots on the second machine. By making use of this fact, they simplify the problem and obtain following formulation with a convex objective function and fewer number of variables and constraints,

$$\begin{array}{lll}
\min F(L) = & p_1 L_1 + p_2 \sum_{k=1}^{s} \sum_{l=1}^{k} L_l L_k & \\
\text{subject to} & & \\
& \sum_{k=1}^{s} L_k = 1 & \\
& p_2 L_k - p_1 L_{k+1} \geq 0 & k = 1, \ldots, s-1 \\
& L_k \geq 0 & k = 1, \ldots, s
\end{array}$$

They, later, observe an interesting characteristic of optimal solution. The size of sublots follow the geometric pattern up to a sublot l, $1 \leq l \leq s$, and the remaining sublots are of the same size, that is

$$\begin{array}{lll}
L_k = & \pi^{k-1} L_1 & k = 1, \ldots, l \\
L_k = & \frac{1 - L_1 \sum_{t=1}^{l} \pi^{t-1}}{s-k} & k = l+1, \ldots, s \\
& \pi L_l \geq L_{l+1} \geq L_l &
\end{array}$$

where

$$\pi = \frac{p_2}{p_1}$$

$\sum_{k=1}^{s} L_k = 1$ implies

$$L_1 = \frac{\frac{\pi^k - 1}{\pi - 1}\pi - (s-k)}{\frac{\pi^{2k}-1}{\pi^2-1}(s-k)\pi + \left(\frac{\pi^k-1}{\pi-1}\right)^2 \pi}$$

They prove the optimality of this pattern of solutions using Lagrangian multipliers. The immediate result of these observations is the following algorithm of $O(s)$ complexity.

```
optimal ← false
$l \leftarrow 1$
while not optimal do
  begin
    $L_1 \leftarrow \frac{\frac{\pi^k-1}{\pi-1}\pi-(s-k)}{\frac{\pi^{2k}-1}{\pi^2-1}(s-k)\pi+\left(\frac{\pi^k-1}{\pi-1}\right)^2\pi}$
    $L_{l+1} \leftarrow \pi^{l-1}L_1$
    $L_{l+1} \leftarrow \frac{1-L_1\sum_{t=1}^{l}\pi^{t-1}}{s-l}$
    if $\pi L_l \geq L_{l+1} \geq L_l$ then
      optimal ← true
  end.
```

The similar results are conjectured by Çetinkaya & Gupta [9], without proving the optimality of the solution.

When the second machine is dominating, in the optimal solution sublots are unequal. But in some real settings, the equal sublots strategy might be more attractive due to design of handling equipment or capability of the system in tracking the number of items within a sublot. It is shown in [26] that equal sublots perform reasonably well in the two machine flow shop. The equal sublot solution is not more than 114 % of the consistent optimal solution.

2.4 Item Completion Time Criterion

The item completion time criterion is relevant when the material handling availability is infinite at the last machine, such as belt conveyors, etc. But we still assume that an item waits the completion of the sublot it belongs in the other machines. As it will become clear, the item completion time criterion poses most of the properties that the sublot completion time has.

For this criterion, the relevant literature is limited. For item completion time criterion, Çetinkaya & Gupta [9] show that when the first machine has the largest processing time the equal sublots solution is optimal. When there are two machines, they prove that the geometric sublots give the best consistent solution. They also give $O(m^2)$+*bi-section search* algorithm for the two-sublot m-machine problem. The algorithm first sets bounds on the optimal solution using the optimal solution of sublot completion time criterion, then converges on the optimal solution by bi-section search. The algorithms we present here find the exact solution in $O(m^2)$ time.

2.4.1 The Model

Under the item completion time criterion, the formulation presented in the previous section for the sublot completion time model is slightly modified. The sublot k starts its processing on the last machine at time $C_{mk} - p_m L_k$. Therefore the t^{th} item in the sublot completes its processing on the last stage at time $C_{mk} - p_m L_k + p_m t$ As it is illustrated in the Figure 2.14, the total flow time is

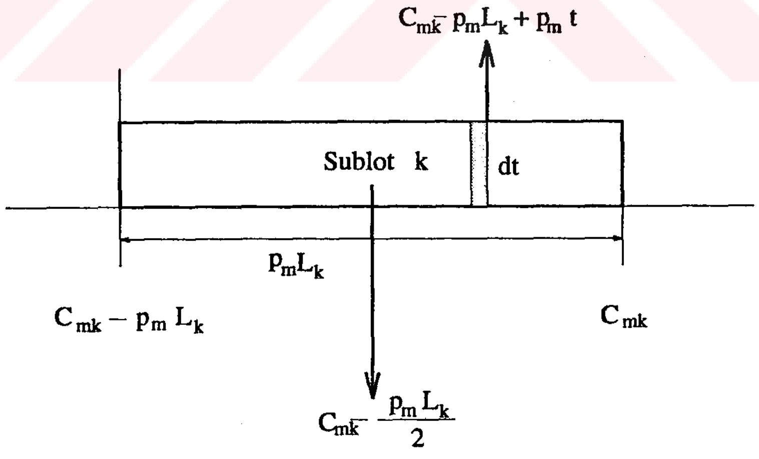

Figure 2.14: Delivery at the last stage in item completion time criterion

$$F(L) = \sum_{k=1}^{s} \left[\int_0^{L_k} (C_{mk} - p_m L_k + p_m t)\, dt \right]$$

$$= \sum_{k=1}^{s} \left[(C_{mk} - p_m L_k)\, t + \frac{p_m t^2}{2} \Big|_0^{L_k} \right]$$
$$= \sum_{k=1}^{s} \left[C_{mk} L_k - \frac{p_m L_k^2}{2} \right]$$

Hence, $-\frac{p_m L_k^2}{2}$, $k = 1, \ldots, s$ terms are the only difference between the item completion and the sublot completion time models. Clearly the item completion time criterion is, in a sense, a relaxed version of sublot completion time criterion, therefore its optimal solution dominates the optimal solution of sublot completion time criterion.

2.4.2 General Results

For the item completion time criterion, when the first machine has the largest processing time, it is shown in [9] that, the equal sublots give the best consistent sublot solution. There is at least one optimal solution where the sublot sizes are in non-decreasing order. By using this fact, objective function of the problem can be reduced to a constant term plus the objective function of the problem under sublot completion time criterion. Therefore the Result 2.4 ,i.e

$$L_k = \frac{1}{s} \qquad k = 1, \ldots, s$$

also holds for item completion time criterion.

2.4.3 Two-Sublot Problem

The objective function of the item completion time model is,

$$\min \quad L_1(C_{m1} - \frac{p_m L_1}{2}) + L_2(C_{m2} - \frac{p_m L_2}{2})$$

using the fact that $L_1 + L_2 = 1$, we get

$$\min \quad C_{m1} L_1 + C_{m2} L_2 - p_m L_1^2 + p_m L_1 - \frac{p_m}{2}$$

Using the similar reductions as in the Section 2.3.3 for $C_{m1}L_1 + C_{m2}L_2$, the following equivalent formulation for the item completion time model is obtained.

$$\min F(L_1) \tag{2.21}$$

subject to

$$\begin{aligned}[(2\textstyle\sum_{l=i}^{m} p_l) - p_i - p_m]\, L_1^2 + \left[p_m + \textstyle\sum_{l=1}^{i} p_l - 2\textstyle\sum_{l=i}^{m} p_l\right] L_1 + \\ \textstyle\sum_{l=i}^{m} p_l - p_m/2 - F(L_1) \leq 0,\end{aligned} \tag{2.22}$$

$$i = 1, \ldots, m$$

The Analysis of the Model

Let a_i, b_i, c_i, again denote the coefficients of L_1^2, L_1, and the constant term. Since $a_i \geq 0$, $i = 1, \ldots, m$, the feasible region of the above problem is also convex. Since we have a linear objective function, the optimum solution is at the boundary of the feasible region. As it is the case in sublot completion time problem, there are finite number of candidate points for the optimum solution. Namely, the intersection points of the parabolas, and the minimum point of each parabola.

The similarities and differences in the problems are outlined in the following two observations.

Observation 9 *The candidate optimal points of the feasible set formed by intersections of the parabolas are the same for both the sublot completion time and the item completion time models. This follows from the fact that the extra terms in the item completion time formulation is independent of index i. Hence the coefficients of joint parabola formed by parabolas t and r,*

$$a_{tr} = a_t - a_r$$

$$b_{tr} = b_t - b_r$$

$$c_{tr} = c_t - c_r$$

are the same in both formulations. Therefore, the parabolas t and r intersect at points $L_1 = 1$, and $L_1 = \frac{c_t - c_r}{a_t - a_r}$.

Observation 10 *The candidate points formed by minimum points of the each individual parabola are,*

$$L_1^{i*} = -\frac{b_i}{2a_i} = \frac{\left[(2\sum_{l=i}^{m} p_l - \sum_{l=1}^{i} p_l) - p_m\right]}{2\left[(2\sum_{l=i+1}^{m} p_l) - p_i - p_m\right]}$$

$$L_1^{i*} = \frac{1}{2} - \frac{\sum_{l=1}^{i-1} p_l}{2\left[(2\sum_{l=i+1}^{m} p_l) + p_i - p_m\right]}$$

Because of the $(-p_m)$ terms in the denominator of the above expression, the minimal values of the parabolas in the item completion time criterion occur at smaller L_1 values than the sublot completion time model.

It should be seen that the Observations 5 through 8 that are stated for the sublot completion time formulation, are also valid for the item completion time criterion. Therefore the $O(m^2)$ algorithms given in the previous section are also applicable to the item completion time model. To establish the finite convergence of the algorithms we need to prove following results.

Result 2.12 *The feasible region of the problem (2.21) has at most m different intersection points.*

The proof is along the same lines as the proof of the Result 2.8.

Result 2.13 *A candidate optimal point is visited at most once by the first algorithm.*

The proof is similar to that of Result 2.6.

Result 2.14 *The Algorithm I converges to an optimal solution.*

Proof: The family of parabola has the features stated in Observations 4 and 7 given for the sublot completion time model. Therefore, for the convergence to an optimal solution, it is enough to show that the curve with the largest index has the greatest slope in the interval $[0, 1/2]$.

$$\begin{aligned}\tfrac{dF_r(L_1)}{dL_1} &= 2a_rL_1 + b_r \\ &= 2[(2\textstyle\sum_{l=r}^{m} p_l) - p_r - p_m]L_1 + p_m + \sum_{l=1}^{r} p_l - 2\sum_{l=r}^{m} p_l \\ &= \textstyle\sum_{l=1}^{r} p_l + 2(2L_1 - 1)\sum_{l=r}^{m} p_l - 2L_1p_r + (1 - L_1)p_m\end{aligned}$$

hence,

$$\begin{aligned}\tfrac{dF_{r+1}(L_1)}{dL_1} - \tfrac{dF_r(L_1)}{dL_1} &= p_{r+1} - 2(2L_1 - 1)p_r - 2L_1(p_{r+1} - p_r) \\ &= (1 - 2L_1)p_{r+1} + (2 - 2L_1)p_r \\ &> 0\end{aligned}$$

from the observation that $L_1^* \leq \frac{1}{2}$ □

Result 2.15 *The intervals $[L_l, L_r]$, computed in each iteration of Algorithm II, contains the optimal solution of the two-sublot problem with item completion time criterion, L_1^*.*

The proof is along the same lines as the proof of Result 2.9.

Therefore, the two sublot problem under the item completion time criterion is solvable in $O(m^2)$ time with Algorithms I or II.

2.4.4 Two-Machine Problem

The two machine lot streaming problem is solvable in polynomial time as shown in [9]. As it is the case in sublot completion time criterion, there are two cases possessing different solution characteristics.

$$p_1 \geq p_2$$

$$p_1 < p_2$$

As mentioned in Section 2.4.2 the equal sublots are optimal for the first case. For the second case, the derivation of the optimal solution is easier than the sublot completion time criterion. The minimization of the item completion time turns out to be the sum of the idle time and a constant term on the second stage. This results in geometric sublots solution,

$$L_k = \pi^{k-1}/(\sum_{l=0}^{s-1} \pi^l)$$

$$\pi = \frac{p_2}{p_1},$$

since the idle time on the second machine is minimized by geometric sublots.

Chapter 3

Multi Job Lot Streaming Problems

The lot streaming and sequencing problems generally arise in the same context, namely, in detailed production scheduling. Very few production system produces only a single type of job and in quite a few production system the jobs are indivisible monolithic entities. In most industrial settings the streaming problem is embedded into the sequencing problem. Obviously, simultaneous consideration of both problems can further improve the performance measures.

One of the well known tractable problems in the scheduling theory is the two machine flow shop under makespan criterion, for which Johnson's algorithm is optimal. It turns out that 2-machine n-job streaming-sequencing problem is a special case of this, and hence, it is also polynomially well solvable. In this chapter we consider the integration of sequencing and streaming decisions first in 2-machine and later in m-machine flow shop.

There are a number of papers in literature studying scheduling of jobs where the overlapping of operations are permitted. Çetinkaya & Kayalıgil [8] considers the minimization of makespan in 2-machine flow shop problem. They assume an unlimited material handling capacity between the machines, and therefore allow the unit sized transfer batches. The sequence independent setups are also included in their model. The setups are detached, that is, it is possible to perform the setups without having the job actually at that machine. They show that the sequencing problem is the special case of two-machine flow shop problem with arbitrary time lags, which is introduced by Mitten [18].

Vickson & Alfredson [32] considers two and three stage flow shop problems with equal sized transfer batches. They give an algorithm similar to one proposed by Çetinkaya & Kayalıgil [8] for 2-stage flow shop and the special cases of 3-stage flow shop. They also give a counterexample where the best non-splitting solution is suboptimal for the mean flow time criterion.

Baker [2] studies the same problem and proposes a unified solution to the no setup, attached and detached setup cases. Baker also examines the special versions of m machine problems where all the machines other then two machines are dominated. All of these papers assume unit sized or equal sized transfer batches, no attention is paid for the best allocation of work into transfer batches to minimize the makespan under limited material handling availability case.

When the unit or equal transfer batches are used, it is superfluous to split the lots. However, when the transfer batches are of different sizes, then splitting the jobs can be more attractive, as shown by Potts & Baker [21]. Çetinkaya [10] investigates 2-machine n-job flow shop problem to find the best sequence and best transfer batch sizes to minimize makespan criterion. His model assumes sequence independent detached setups. He further shows that the streaming and sequencing decisions can be given sequentially for the makespan criterion, i.e. first find the transfer batch sizes for each job individually to compute the start and the stop time lags, and then find the best sequence.

Vickson [31] obtains the same results and further he proves that when job splitting is not allowed the lot streaming problem is independent from the sequencing problem for any regular measures of performances. For the continuous sublots problem, he proposes some closed form solutions for computing the transfer batch sizes in the presence of attached and detached setups. For the same problem under integral sublots restriction, he presents polynomial algorithms for both attached and detached setup cases.

3.1 Problem

For two machine flow shop problem involving arbitrary time lags, Mitten [18] shows that, applying Johnson's algorithm to the time lags gives the best *permutation* schedule for the makespan criterion. Let the start and stop lags be denoted as l_j and l'_j, respectively (Figure 3.1). A start lag of length l_j indicates that, the job j cannot start processing on machine two, before at least l_j time units pass upon its completion on machine 1. Similarly a stop lag l'_j means that the job j cannot finish its processing on machine 1 before at least l'_j time units pass upon its completion on the machine one. In Mitten's original work, time lags are permitted to have any value, including negative values. Negative time lags arise in the following two cases. First, when a job can start and finish on the second machine before its completion on the first machine. Second, when there is a long setup on the second machine or a long removal time on the first machine. Following conditions are sufficient for a best permutation schedule to be globally optimal.

$$p_{1j} \geq l_j \geq 0, \quad (\text{or } p_{2j} \geq l'_j \geq 0) \qquad 1 \leq j \leq n$$

In the following section, it is shown that a streamed job satisfies this condition, in two-machine flow shop.

3.2 Two-Machine Problem

Consider a single job to be processed on two machines, moreover assume that the job can be transferred to the next machine using s transfer batches. For any choice of transfer batch sizes the resulting configuration yields a job with two positive time lags. To obtain this configuration, the sublots on the second machine are shifted to the right to obtain continuous production without changing the time to complete the job (Figure 3.1). As it can be seen from the figure the start and stop lag of job j is,

$$\begin{aligned} l &= C_{max} - p_1, \\ l' &= C_{max} - p_2. \end{aligned}$$

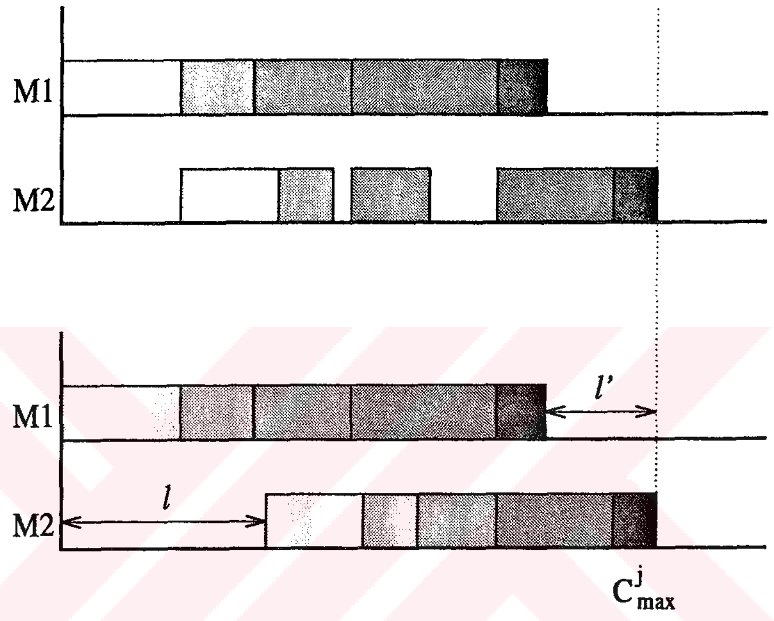

Figure 3.1: Time lags in streaming problem

For a given sublot sizes, the optimal makespan can be found by applying Johnson's algorithm to l_j and l'_j. When the lot streaming and sequencing decisions are considered simultaneously in two machine flowshop, it turns out that the decisions can be made hierarchically to find the best makespan. The following result is a straightforward extension of the one given by Akyel [1]. The result shows that for any regular measure of performance, there is at least one optimal solution in which sublot sizes of each job follows a geometric pattern.

Result 3.1 *For 2-machine n-job sequencing and streaming problem where job j can be transferred to the next machine using s_j transfer batches, there exists* *an optimal solution in which each job is streamed independently using*

$$L_k = \pi^{k-1}/(\sum_{l=0}^{s-1} \pi^l) \tag{3.1}$$

$$\pi = \frac{p_2}{p_1}$$

Proof: Let Π be an optimal sequence of the jobs and L_{jk}, $j = 1 \ldots n$, $k = 1 \ldots s_j$ be the optimal sublot sizes. Assume that for at least one job, the sublot sizes are different from the ones computed from (3.1). Let the job r be one of these jobs. Let $[S_{1r}, C_{1r}]$ and $[S_{2r}, C_{2r}]$ be the time intervals allocated to the job r on machine 1 and machine 2. respectively. Assume that instead of L_{rk}, $k = 1 \ldots s_r$, we use the sublot sizes computed from (3.1). Clearly, for any choice of sublot sizes sizes, the interval $[S_{1r}, C_{1r}]$ can be allocated to the job r on machine 1. Since the sublot sizes computed from (3.1) minimizes the makespan of job r, the new completion time of job r on the second machine can not be greater than C_{2r} (Figure 3.2). Hence $[S_{2r}, C_{2r}]$ time interval can still be allocated to the job r without increasing the optimum value of the sequence Π. Hence, if every job has the sublot sizes computed from (3.1), the performance measure of the sequence Π will not be worse. $\square$

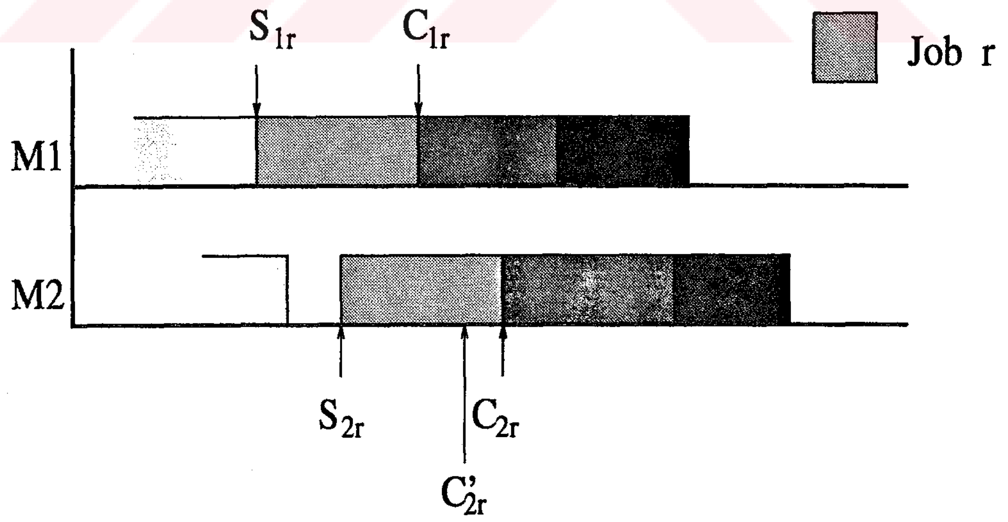

Figure 3.2: n-job lot streaming problem

When a single job is streamed using the geometric pattern (3.1), the start lag is simply processing time of first sublot on machine 1 and the stop lag is the processing time of last sublot on machine 2, (Figure 3.3) formally for a job j we have start lag l and stop lag l' as

$$l_j = p_{1j}\ L_{j1}$$

$$l'_j = p_{2j}\ L_{js}$$

Also observe that

$$l_j \leq p_{1j}$$

since the length of overlapping area is positive for every job. Therefore, the Mitten's results can be used to establish the global optimality.

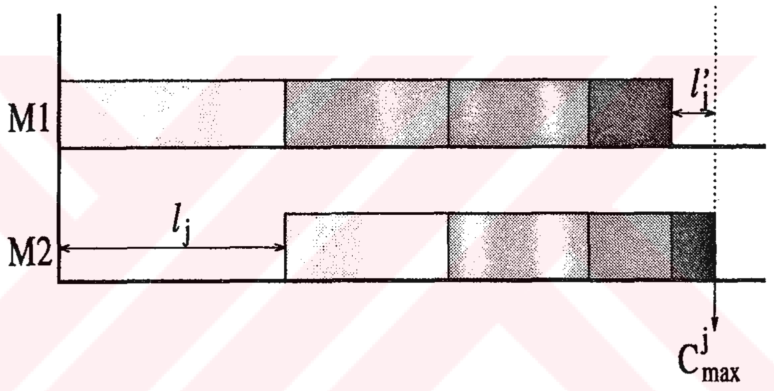

Figure 3.3: Time lags in the geometric solution

The immediate result of Result 3.1 is the following algorithm for the problem of lot streaming and sequencing n-jobs in a two machine flowshop. The algorithm computes the optimal transfer batch sizes and start and stop lags for each job in the "*for*" loops. Later, it finds the sets F and B containing the jobs that are to be sorted according to their start or stop lags. Finally, it concatenate both set to get optimal sequence.

```
for $j = 1$ to $n$ do
  begin
    $\pi_j \leftarrow \frac{p_{2j}}{p_{1j}}$
    $L_{j1} \leftarrow \frac{1}{\sum_{l=0}^{s-1} \pi_j^l}$
    for $k = 2$ to $s_j$ do
      $L_{jk} \leftarrow \pi_j \cdot L_{j,k-1}$
    $l_j \leftarrow p_{1j} \cdot L_{j1}$
    $l'_j \leftarrow p_{2j} \cdot L_{js}$
  end

$F \leftarrow \{j : l_j \leq l'_j\}$
$B \leftarrow \{j : l_j > l'_j\}$

Sequence the jobs in $F$ in the non-decreasing order of $l_j$ to get $\Pi_1$
Sequence the jobs in $B$ in the non-increasing order of $l'_j$ to get $\Pi_2$

The optimal sequence is $\Pi_1 + \Pi_2$
```

3.3 Extensions to the Flow Shops Containing 3 or More Machines

The results of previous section cannot be generalized to the flow shops containing three or more machines. Minimization of makespan in a three stage flow shop is known to be $\mathcal{NP}-Hard$ [23]. Therefore, the three machine problem with lot streaming is expected to be harder to solve than the 2-machine problem. Consequently, in this section, our aim is to detect the cases of m-machine sequencing and lot streaming problems that are polynomially solvable.

Machine dominance results of classical flow shop theory are not directly applicable to the lot streaming problem. The domination of a machine implies that the dominated machine can not be bottleneck for any sequence. An extensive survey of the solvable cases of permutation flow shops are given by Monma & Rinnooy Kan [20]. However, if lot streaming is also allowed, under the sublot sizes of jobs must also be taken into account as well as job processing times. A machine that is non-bottleneck when there is no streaming may turn out to be bottleneck when streaming is permitted. One such example is illustrated in Figure 3.4. There are two jobs which can be transferred to the next machine using two sublots. Clearly when lot streaming is not permitted, the machine two is dominated by machine three. But in the lot streaming problem, the machine two turns out to be the bottleneck machine. If the transfer batches of the jobs are found individually and then jobs are sequenced to find the best makespan, the makespan turns out to be 10.62. But the optimal solution of the problem is 10.50 and the optimal sublot sizes are different from the ones found by streaming each job independently. Therefore, the lot streaming problem is not independent from the sequencing problem for the shops containing more than two machines.

One other difficulty in three or more machines problem is that the streamed problem cannot be directly seen as the time lag problem. The start and stop lags are ambiguous, since there are virtually two start and two stop lags as seen from the Figure 3.5. There is no clear criterion to sequence the jobs. Because of these, we consider the hierarchical application of the lot streaming and sequencing problems. Assume that in an m-stage flow shop we have n jobs to sequence. A heuristically good strategy is to stream each job independently and later sequence these jobs according to the resulting configuration.

In the next result, we try to find the best sequence of jobs for a special case of the m-machine problem to minimize the makespan, given the best sublot sizes minimizing the individual makespan of each job.

Result 3.2 *In the sequencing problem in an m-stage flow shop given the optimal sublots for each individual job, if the following conditions hold, then the*

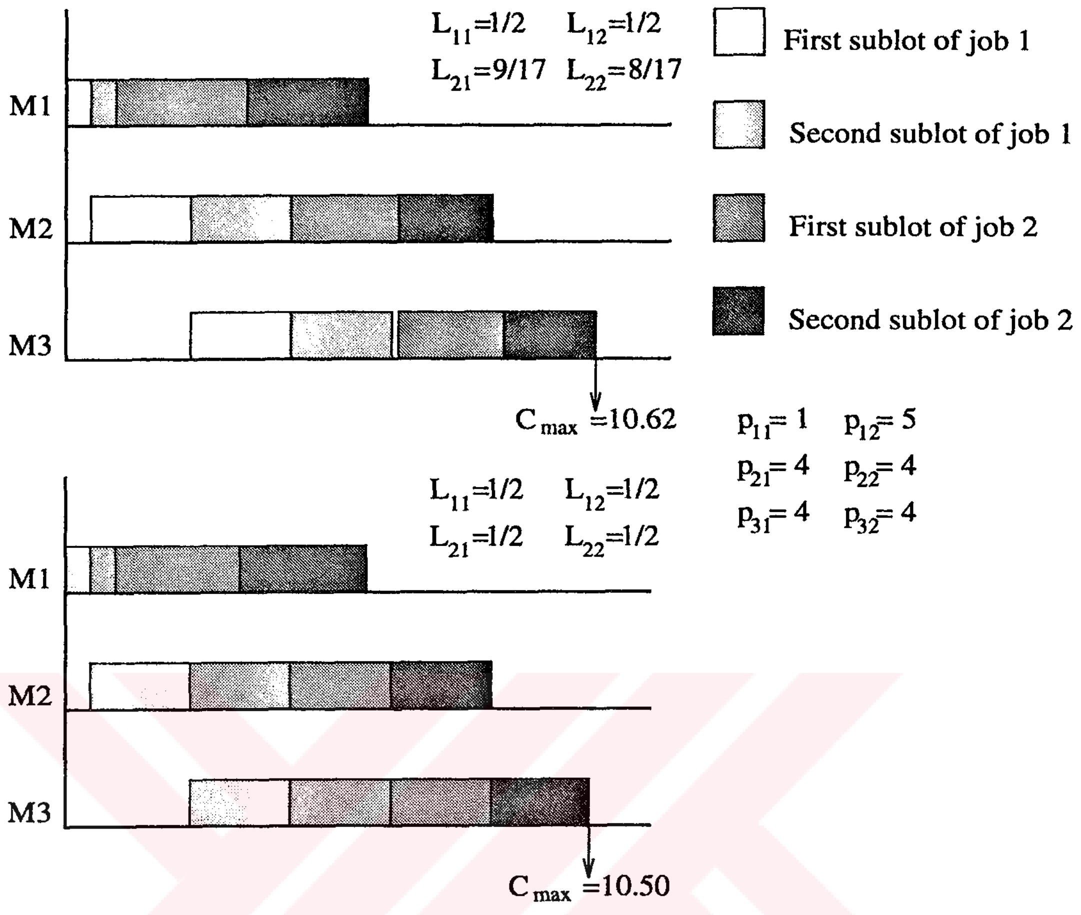

Figure 3.4: Lot streaming with second machine dominated

best schedule can be found by applying the Johnson's algorithm.

$$p_{1j}\, p_{mj} \geq (\sum_{i=2}^{m-1} p_{ij})^2 \quad 1 \leq j \leq n. \tag{3.2}$$

$$\max_{\{1\leq j\leq n\}}\{\sum_{i=2}^{m-1} p_{ij}\} \leq \min_{\{1\leq j\leq n\}}\{p_{1j}\}. \tag{3.3}$$

$$p_{1j} \geq p_{mj}. \tag{3.4}$$

Sublots are consistent and each job has s sublots (3.5)

The optimal lot streaming strategy for a single job in a shop satisfying the condition (3.2), was given in the Section 2.2.5 as,

$$L_k = \frac{\pi^{k-1}}{\sum_{l=0}^{s-1} \pi^l} \tag{3.6}$$

where

$$\pi = \frac{\sum_{i=2}^{m} p_i}{\sum_{i=1}^{m-1} p_i}. \tag{3.7}$$

This solution has some interesting characteristics. There are s distinct critical paths, each of which contains a distinct sublot as the critical sublot (Figure 3.5). Also each critical path contains only one sublot on the interior machines.

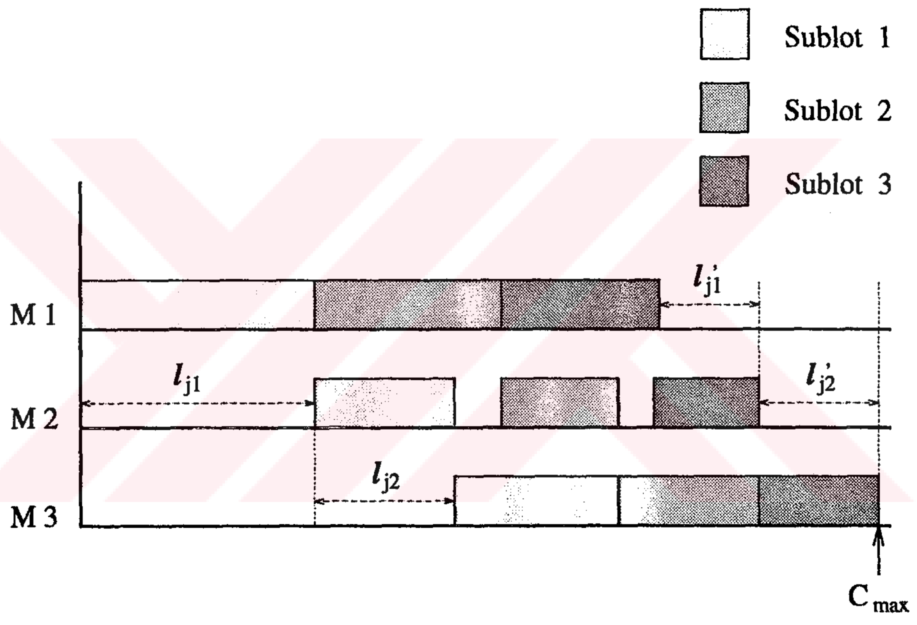

Figure 3.5: Time lags in m-machine flow shop

The following notation will be used in the presentation of the subsequent results (Figure 3.5),

$$\begin{aligned} l_{j1} &= L_{j1}\, p_{1j} \\ l_{j2} &= L_{j1} \sum_{i=2}^{m-1} p_{ij} \\ l'_{j1} &= L_{js} \sum_{i=2}^{m-1} p_{ij} \end{aligned}$$

$$\begin{aligned} l'_{j2} &= L_{js}\, p_{mj} \\ l_j &= l_{j1} + l_{j2} \\ l'_j &= l'_{j1} + l'_{j2} \end{aligned}$$

As proven in Section 2.2.1, when the first machine has the largest processing time, if each job is streamed individually, the optimal sublot sizes are in non-increasing order for each job, that is,

$$j \geq k \Rightarrow L_j \geq L_k, \tag{3.8}$$

We first prove two results to show that, the second machine cannot be a bottleneck machine for any sequence of the jobs.

Result 3.3 *If the conditions (3.2), (3.3), (3.4) and (3.5) hold then,*

$$\min_{\{1 \leq j \leq n\}} \{l_j\} \geq \max_{\{1 \leq j \leq n\}} \{l'_j\} \tag{3.9}$$

Proof: Take any two jobs t and r arbitrarily. Observe that

$$l_{t1} = L_{t1}\, p_{1t} \quad \text{and} \quad l'_{r1} = L_{rs} \sum_{i-2}^{m-1} p_{ij}.$$

Since $L_{t1} \geq L_{t2} \geq \ldots \geq L_{ts}$ and $\sum_{k=1}^{s} L_{tk} = 1$, we have

$$L_{t1} \geq \frac{1}{s},$$

and also $L_{r1} \geq L_{r2} \geq \ldots \geq L_{rs}$ and $\sum_{k=1}^{s} L_{rk} = 1$, implies

$$L_{rs} \leq \frac{1}{s},$$

From assumption (3.3) we have

$$p_{1t} \geq \sum_{i=2}^{m-1} p_{ir}$$

Thus

$$l_{t1} = p_{1t}\ L_{t1} \geq \frac{1}{s}\ p_{1t} \geq \frac{1}{s}\sum_{i=2}^{m-1} p_{ir} L_{rs} = l'_{r1} \qquad \square$$

Result 3.4 *If each job streamed individually using (3.6) then, for any sequence of the jobs, the critical path defining the makespan of the schedule can not contain more than one sublot on the second machine.*

Proof: Assume that, there exists a sequence Π in which there are more than two operations on the critical path defining the makespan. To simplify the notation, assume that

$$(\Pi(1), \Pi(2), \ldots, \Pi(n)) = (1, 2, \ldots, n)$$

Let c be the index of the job whose operation on the second machine is the first of these critical operations. The case is illustrated in Figure 3.6. As seen from the figure, for the job $c + 1$ to be critical on one of the interior machines, the following condition should be satisfied,

$$l'_{c1} \geq l_{c+1,1},$$

but this contradicts with the Result 3.3.

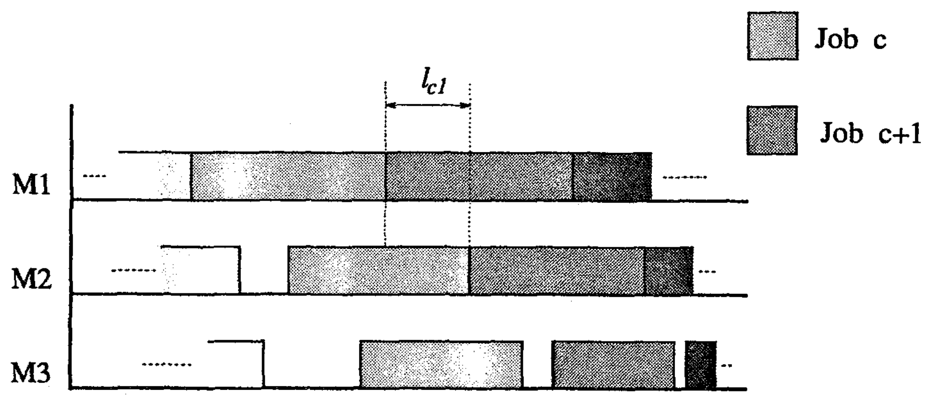

Figure 3.6: First critical sublot on the second machine

Proof [Result 3.2]: As it is shown in Result 3.4 (Figure 3.6), there cannot be any interference of sublots of two different jobs on the interior machines. Hence, the makespan $C_{max}(\Pi)$ for a sequence Π is,

$$C_{max}(\Pi) = \left\{ \sum_{j=1}^{c-1} p_{1\Pi(j)} + C_{max}^{\Pi(c)} + \sum_{j=c+1}^{n} p_{3\Pi(j)} \right\}$$

where c is the critical job in the sequence Π. $C_{max}^{\Pi(j)}$ is the makespan of the job j, when it is streamed individually.

Then the minimum makespan is

$$C_{max} = \min_{\Pi} \left\{ \sum_{j=1}^{c-1} p_{1\Pi(j)} + C_{max}^{\Pi(c)} + \sum_{j=c+1}^{n} p_{3\Pi(j)} \right\}$$

Let O_j denote the length of overlap in between the operations of job j on machine one and three. Clearly,

$$O_{\Pi(c)} = p_{1\Pi(c)} + p_{3\Pi(c)} - C_{max}^{\Pi(c)}$$

$$= \min_{\Pi} \left\{ \sum_{j=1}^{c-1} p_{1\Pi(j)} + p_{1\Pi(c)} + p_{3\Pi(c)} - O_{\Pi(c)} + \sum_{j=c+1}^{n} p_{3\Pi(j)} \right\}$$

$$= \min_{\Pi} \left\{ \sum_{j=1}^{c} p_{1\Pi(j)} - O_{\Pi(c)} - \sum_{j=1}^{n} O_{\Pi(j)} + \sum_{j=c}^{n} p_{3\Pi(j)} \right\} + \sum_{j=1}^{n} O_{\Pi(j)}$$

$$= \min_{\Pi} \left\{ \sum_{j=1}^{c} \left(p_{1\Pi(j)} - O_{\Pi(j)}\right) + \sum_{j=c}^{n} \left(p_{3\Pi(j)} - O_{\Pi(j)}\right) \right\} + \sum_{j=1}^{n} O_j$$

$$= \min_{\Pi} \left\{ \sum_{j=1}^{c} l_{\Pi(j)} + \sum_{j=c}^{n} l'_{\Pi(j)} \right\} + \sum_{j=1}^{n} O_j$$

Since the second term is a constant, the last expression is equivalent to

$$\min_{\Pi}\left\{ \sum_{j=1}^{c} l_{\Pi(j)} + \sum_{j=c}^{n} l'_{\Pi(j)} \right\}$$

But the last expression is the objective function of the two-machine flow shop problem, with p_j's are replaced by l_j, for which the Johnson's algorithm is optimal. □

Observe that

$$l'_j = L_s \sum_{i=2}^{m} p_i \le L_1 \sum_{i=1}^{m-1} p_i = l_j$$

Therefore, sorting the jobs in non-decreasing order of their stop lags, gives the best sequence.

Now if we replace the conditions (3.3) & (3.4) with conditions

$$\max_{\{1\le j\le n\}}\{\sum_{i=2}^{m-1} p_{ij}\} \le \min_{\{1\le j\le n\}}\{p_{1j}\}. \quad (3.10)$$

$$p_{mj} \ge p_{1j}. \quad (3.11)$$

Because of the symmetry, now we have

$$\max_{\{1\le j\le n\}} l_{j1} \le \min_{\{1\le j\le n\}} l'_{j1} \quad \text{and} \quad l_j \le l'_j$$

Hence, in this case the optimal schedule can be found by sorting jobs in non-increasing order of their start lags.

Chapter 4

Conclusions and Further Research

The purpose of the this study was to investigate the various forms of lot streaming problem to derive the basic solution characteristics and identify the well solvable cases. For this purpose, we first studied single job lot streaming problems, later extended our results to multi-job problems.

For the single job lot streaming problem, we considered three measures of performance, sublot, item and job completion time criteria. Under job completion time criterion, we derived an exact solution scheme for a special case of m-machine s-sublot problem satisfying,

$$p_1 p_m > \left(\sum_{i=2}^{m-1} p_i\right)^2$$

For the sublot completion time criterion, when the first machine has the largest processing time, we derived the exact solution scheme. In order to establish this result, we first showed the existence of an optimal solution in which the sublot sizes are in increasing order. By making use of this fact, we proposed a simplified formulation containing fewer number of constraints with a convex objective function. Later we derived the optimal solution using Lagrange multipliers method.

We considered the two-sublot problem. By making use of some results from the scheduling theory, we transformed the standard formulation to two variable minimax optimization problem with quadratic constraints. After investigating the characteristics of the feasible set, we proposed two algorithms each having $O(m^2)$ complexity, where m denotes the number of machines. This bound is equal to the one proposed by Çetinkaya and Gupta [9]. In order to compare the algorithms we carried out an experimental study. The empirical running time of the algorithms turned out to be much better than the their worst case behavior. For two-machine problem under sublot completion time criterion, we gave an example to show the suboptimality of the consistent sublot solutions.

In order to present the results pertaining to item completion time criterion, we pointed out the similarities in between the sublot and item completion time criteria. Later, we showed that $O(m^2)$ algorithms given for the sublot completion time criterion are also applicable to the item completion time criterion. This is the best bound for this criterion.

Finally, we attacked the multi-job streaming problems. For two machine problem, we gave a simpler alternative derivation of the Vickson's [31] results on the independence of the streaming and sequencing problem. We also gave a three machine example in which the second machine is dominated in classical scheduling terms, to show that the lot streaming and sequencing problems are coupled. We also investigated performance of the hierarchical application of streaming and sequencing decisions. For a special case of this problem, we showed that the sequencing problem reduces to two machine problem.

There are several directions for the further research. The experimental study of the algorithms for the item and sublot completion time criteria, pointed out the existence of the algorithms with better bound. By investigation of further characteristics of the feasible set of problem, further reduction in these bounds is likely.

Due to the its nonlinear objective function, the general m-machine s-sublot problem under consistency assumption cannot be adressed for sublot and item completion time criteria. The solution properties of the two-machine and two-sublot problems can be used to devise heuristics for the m-machine s-sublot problems.

In our experimental studies, for the three-machine problem when one or two machines are dominated, we observed several structural patterns in the optimal solutions, indicating the possible existence of exact solution algorithms.

The variable sublot lot streaming problems can be investigated by studying the formulation of the general lot streaming problem given by Benli [6]. Although his formulation is inherently complicated, for two sublot and two machine problems exact algorithms may be driven.

The multi-job lot streaming problem is a promising area for the further research. Although we showed that the classical dominance results does not apply for the lot streaming problem, our empirical studies pointed the existence of more restricted special cases, such as

$$\min_j\{p_{1j}\} \geq \max_j\{p_{2j}\} \;\;\&\;\; \min_j\{p_{2j}\} \geq \max_j\{p_{3j}\}$$

or,

$$\min_j\{p_{1j}\} \geq \max_j\{p_{2j}\} \;\;\&\;\; \min_j\{p_{3j}\} \geq \max_j\{p_{2j}\}$$

where the streaming problem can be decoupled from the sequencing problem.

Considering streaming and sequencing problems hierarchically is a good strategy, when the simultaneous consideration complicates the problem. In the three machine problem, when the condition

$$p_{1j}\, p_{3j} \geq (p_{2j})^2 \qquad 1 \leq j \leq n$$

is relaxed, we observed that the optimal sequence could still be found by applying the Johnson's algorithm. But the proof of

$$\min_{\{1\leq j\leq n\}}\{l_j\} \geq \max_{\{1\leq j\leq n\}}\{l'_j\}$$

for the case

$$p_{1j}\, p_{3j} < (p_{2j})^2 \qquad 1 \leq j \leq n.$$

is not available. Hence we give this result as a conjecture.

Conjecture 1 *In the problem of lot streaming and sequencing in the 3-stage flow shop problem, if the following conditions hold, then the best schedule can be found by applying Johnson's algorithm.*

$$\begin{aligned} \max_{\{1\leq j\leq n\}}\{p_{2j}\} &\leq \min_{\{1\leq j\leq n\}}\{p_{1j}\}. \\ p_{1j} &\geq p_{3j}. \end{aligned}$$

Sublots are consistent and each job has s sublots

Investigation of other special cases along with the proof of this result, can be a promising direction for further research.

Bibliography

[1] Cemal Akyel. Private communication.

[2] K. R. Baker. *Elements of Sequencing and Scheduling.* The Amos School of Business Administration, Dartmouth College, Hanover, N.H., 1994.

[3] K. R. Baker and D. Jia. A comparative study of lot streaming procedures. *OMEGA Int. J. of Mgmt. Sci.*, 21:561–566, 1993.

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

[5] Kenneth R. Baker and David F. Pyke. Solution procedures for the lot-streaming problem. *Decision Sciences*, 21:475–491, 1990.

[6] Ömer Benli. Streaming a single job in a flow shop. Research Report IEOR-9409, Department of Industrial Engineering, Bilkent University, 1994.

[7] Jimmie Browne, John Harhen, and James Shivnan. *Production Management Systems: A CIM Perspective.* Addison-Wesley, 1988.

[8] F. Can Çetinkaya and M. Sinan Kayalıgil. Unit sized transfer batch scheduling with setup times. *Computers and Industrial Engineering*, 22:177–183, 1992.

[9] Ferda C. Çetinkaya and Jatinder N. D. Gupta. Flowshop lot streaming to minimize total weighted flow time. Research Memorandum 94-24, Industrial Engineering, Purdue University, 1994.

[10] Ferda Can Çetinkaya. Lot streaming in a two-stage flow shop with set-up, processing and removal times separated. *J. Opl Res. Soc.*, 45, 1994. forthcoming.

[11] Richard W. Conway, William L. Maxwell, and Louis W. Miller. *Theory of Scheduling.* Addison-Wesley, 1967.

[12] R. A. Dudek, S. S. Panwalkar, and M. L. Smith. The lessons of flowshop scheduling research. *Ops Res.*, 40:7–13, 1992.

[13] C. A. Glass, J. N. D. Gupta, and C. N. Potts. Lot streaming in three-stage production processes. *European J. Operations Research*, 75:378–394, 1994.

[14] S. K. Goyal. Note on "manufacturing cycle time determination for a multi-stage economic production quantity model. *Mgmt Sci.*, 23:332–333, 1976.

[15] S. M. Johnson. Optimal two and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1:61–68, 1954.

[16] D. H. Kropp and T. L. Smunt. Optimal and heuristic models for lot splitting in a flow shop. *Decision Sciences*, 21:691–709, 1990.

[17] R. Lundrigan. What is this thing called OPT? *Prod. and Inv. Mgmt.*, 27:2–12, 1986.

[18] L. G. Mitten. Sequencing *n* jobs on two machines with arbitrary time lags. *Mgmt Sci.*, 5:293–298, 1959.

[19] J. P. Moily. Optimal and heuristic procedures for component lot-splitting in multi-stage manufacturing systems. *Mgmt Sci.*, 32:113–125, 1986.

[20] C. L. Monma and A. H. G. Rinnooy Kan. A concise survey of efficiently solvable special cases of the permutation flow-shop problem. *Recherche operationnelle*, 17:105–119, 1983.

[21] C. N. Potts and K. R. Baker. Flow shop scheduling with lot streaming. *Opns. Res. Lett.*, 8:297–303, 1989.

[22] C. N. Potts and L. van Wassenhove. Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. *J. Opl Res. Soc.*, 48:395–406, 1992.

[23] A. H. G. Rinnooy Kan. *Machine Scheduling Problems: Classification, Complexity, and Computations.* Martinus Nijhoff, The Hague, 1976.

[24] M.L. Smith, S.S. Panwalkar, and R.A. Dudek. Flowshop sequencing problem with ordered processing time matrices: A general case. *Mgmt Sci.*, 21:544–549, 1975.

[25] A. Z. Szendrovits. Manufacturing cycle time determination for a multi-stage economic production quantity model. *Mgmt Sci.*, 22:298–308, 1975.

[26] Alper Şen, Engin Topaloğlu, and Ömer Benli. Optimal streaming of a single job in a two stage flow shop. Research Report IEOR-9413, Department of Industrial Engineering, Bilkent University, 1994.

[27] Engin Topaloğlu, Alper Şen, and Ömer Benli. Optimal streaming of a single job in a flow shop with two sublots. Research Report IEOR-9412, Department of Industrial Engineering, Bilkent University, 1994.

[28] 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.

[29] Dan Trietsch and Kenneth R. Baker. Basic techniques for lot streaming. *Ops Res.*, 41:1065–1076, 1993.

[30] M. Michael Umble and M. L. Srikanth. *Synchronous Manufacturing: Principles For World Class Excellence.* APICS South-Western series in production and operations management. South-Western Publishing, Cincinnati, Ohio, 1990.

[31] R. G. Vickson. Optimal lot streaming for multiple products in a two-machine flow shop. *European J. Operations Research*, 1994. forthcoming.

[32] R. G. Vickson and B. E. Alfredsson. Two- and three-machine flow shop scheduling problems with equal sized transfer batches. *Int. J. Prodn Res.*, 30:1551–1574, 1992.

[33] T. Vollmann. OPT as an enhancement to MRPII. *Prod. and Inv. Mgmt.*, 27:38–46, 1986.

[34] E. F. Williams and S. Tufekci. Polynomial time algorithms for the $m \times 2$ lot streaming problem. Research Report 92-10, Department of Industrial and Systems Engineering, University of Florida, 1992.