# Lot Streaming in Hybrid Flow Shop Scheduling

**Masood Ghasemi**

**A Thesis**

**in**

**The Department**

**of**

**Mechanical and Industrial Engineering**

**Presented in Partial Fulfillment of the Requirements**

for the Degree of Master of Applied Science (Industrial Engineering) at

Concordia University

Montreal, Quebec, Canada

**May 2008**

*Your file Votre référence*
*ISBN: 978-0-494-45298-1*
*Our file Notre référence*
*ISBN: 978-0-494-45298-1*

# Abstract

Production planning and scheduling play significant roles in manufacturing system operations and different techniques have been used to enhance their performance. Lot streaming has been studied for decades and is shown to accelerate production flow. This research deals with lot streaming in hybrid flow shops. Multiple products are processed in a multi-stage hybrid flow shop with non-identical machines. Sublots can be constant or consistent and intermingling is not allowed. Setups are attached and sequence independent. The problem is to simultaneously determine product sequence and sublots sizes so that the makespan is minimized. The model presented in this thesis is a mixed integer linear programming formulation for solving this problem. Several variations of the model are presented to incorporate different problem settings such as exploitation of variable sublots in the single product problem. Numerical examples are presented to validate the proposed model and to compare it to similar example problems in the literature. Furthermore, an example of a lot streaming problem in a general multi-stage hybrid flow shop is concerned and discussions and analysis are presented.



Keywords: *Production planning; Scheduling; Lot streaming; Hybrid flow shop; Integer programming*

# Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Mingyuan Chen, for his support and guidance in my thesis. His encouragement and useful comments have enabled me to reach this education level.

I would like to appreciate Dr. M. Defersha, my friends, and my colleagues for their precious comments and critiques in fulfillment of my thesis.

I profoundly thank my parents, my sisters and my brother. Without their love and encouragement, this work would not have been possible.

# Table of Contents

# List of Figures

# List of Tables

# Chapter One Introduction

## 1.1 Foreword

In today global manufacturing competition, several strategies are considered by managers to improve company's performance. Although many functions can be optimized in a company, production decisions directly affect the manufacturing performance. Customers' needs and expectations are increasing rapidly and companies compete to better meet their requirements; therefore, it is a success key to make the right product at the right time, to shorten production life cycles and to lower production costs.

## 1.2 Production planning and scheduling

Planning is crucial in production which is often complex due to occurrence of unexpected events in manufacturing environment. It is the coordination of activities and resources over time to optimize resource allocation and to achieve organization's goals. Production planning is facilitated by various techniques such as manufacturing resource planning (MRP II) and just-in-time (JIT) which are most popular methods. Nervousness and infeasible schedules which lead to high work in process and dissatisfaction of due dates, are some characteristics of MRP II (Biskup *et al.*, 2006). Considerable research work is done to improve inefficiency of these systems. Modifying and rearranging the floor schedules which mean adding, resizing, or replacing the existing lots have been regarded to be much effective.

Scheduling, workforce planning, facility planning, and cost planning are four elements of manufacturing planning. Scheduling is allocating limited resources to tasks to optimize objectives and to satisfy customers' demand with on-time delivery and has an important role in total quality management (Lee et al. 1997). Production planning and scheduling should be often incorporated in a single structure to manage manufacturing processes from getting the order until shipping the products.

Most production systems involve multiple products that are processed by multiple production stages, having inventories in between and utilizing different means of transportation. These operations are very complex since the system should decide how to allocate jobs to machines. As such, scheduling problems are optimization problems where the objective can be developing a job allocation so demand is satisfied, costs are minimized and profit is maximized. Scheduling becomes more complex in dynamic and stochastic production environments with unpredictable demand and stochastic factory output.

## 1.3 Types of scheduling problems

Based on certain characteristics, scheduling problems are categorized and studied by researchers in different types.

**Flow shop scheduling**

In this type of scheduling problem, jobs are processed in more than one stage. Each machine processes one job at a time and all jobs must be processed by all the machines in the same order; therefore, number of operations of a job is equal to the number of machines. This scheduling problem, except certain simple cases, is NP-hard and complexity increases with number of machines and jobs.

**Hybrid flow shop (HFS) scheduling**

Hybrid flow shop scheduling is the generalization of flow shop scheduling problems where a stage may have multiple processing machines. Researchers have been interested in this type of scheduling because of the nature of relevance to manufacturing and computer systems. This type of problem is concerned in situations where average processing times of jobs in some stages are rather high. Parallel machines can be an alternative in these stages to accelerate production rate and reduce WIP by eliminating or smoothing bottle necks. Flexible flow shop (FFS) scheduling is a type of hybrid flow shop scheduling where jobs may skip some stages so it is also called hybrid flexible flow shop problem.

**Job shop scheduling**

Job shop scheduling is different from flowshop scheduling in that each job may have specific processing order and might not necessarily meet all the stages. A job may require multiple processing on a single machine. Open shop scheduling is a type of job shop scheduling when the processing order is arbitrary. The solution to this type of scheduling problem comprises total order of operations of a job as well as total order of operations on a machine.

## 1.4 Objectives of scheduling problems

- Minimizing makespan

Makespan or total completion time is the amount of time required for jobs/products to complete processing on certain sets of machines. This objective is concerned wherever the emphasis is on increasing the production rate and meeting the demand. By minimizing makespan in scheduling problems production flow is accelerated and more jobs may be produced in less amount of time.

- Minimizing mean weighted tardiness

Tardiness is the amount of time a job finishes processing after its due date. Minimizing the mean weighted (by job priority) tardiness is an objective in multi-job scheduling problems. Considering penalty on late delivery, this objective is considered to minimize costs and satisfy customers.

- Minimizing mean weighted earliness

Jobs may finish processing earlier than their due date. In lean production systems one of the goals is to produce jobs on time; any early delivery has inventory holding cost. Therefore, in multiple-job scheduling problems minimizing mean weighted earliness is sometimes considered as an objective.

- Minimizing maximum tardiness / earliness

It is another goal in scheduling problems which deals with number of tardy/early jobs rather than the time. Considering this objective function, a schedule of jobs is achieved to minimize maximum tardiness / earliness or to minimize number of tardy / early jobs.

Based on the problem characteristics, each of these objectives is considered in scheduling problems. Some of them may also be combined together in a single structure to cover different requirements of a company.

## 1.5 Lot streaming

Lot streaming, introduced by Reiter (1966), is a method to split a production lot into sublots and then scheduling sublots on machines in order to accelerate the process of a job in production line. In production systems, without lot streaming, the whole lot is transferred to the next stage in its schedule with a fixed size; when processing a part is finished on a machine it has to wait in the output buffer until the whole lot is completed whereas the successive machine might be idle. By splitting the lots into sublots and overlapping processes, the next machine starts processing even though its predecessor machine has not finished the whole lot. Most researchers consider number and size of sublots as decision variables in lot streaming problems. Lot streaming is an alternative which improves schedules and assignments and facilitates shop floor decision making. Moreover, this technique is less costly and less time consuming because it can be implemented in the current production line without any need to change facilities and production processes.

### 1.5.1 Lot streaming categories

Chang and Chiu (2004) divide existing lot streaming problem into four categories of single product, multiple-product, time-related, and cost-related problems. As illustrated in Figure 1-1, these categories are based on number of products and performance measurements in lot streaming problems. Few research works consider both lot-sizing and sequencing issues to optimize cost and time together. Potts and Van Wassenhove (1992) study the interaction of time and cost and allow trade-offs in integrating batching, lot-sizing and scheduling in a complex environment.

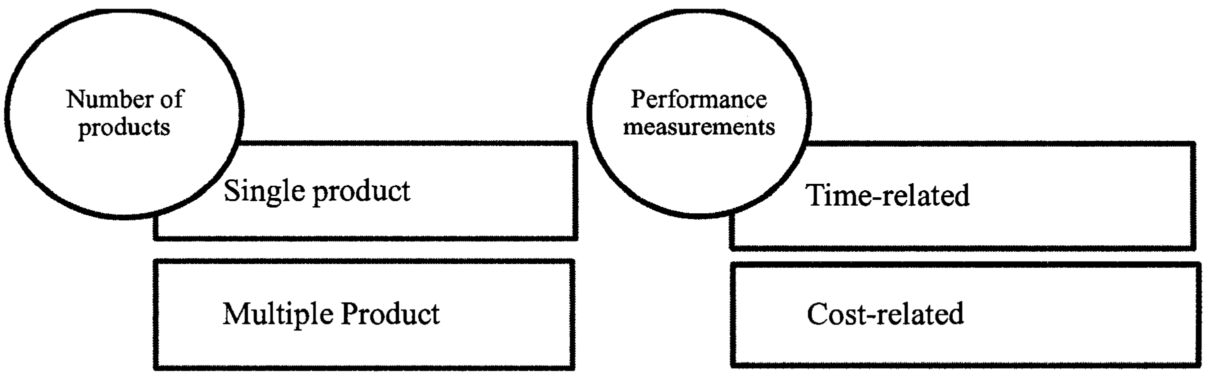



**Figure 1-1 Four categories of lot streaming problems**

### 1.5.2 Lot streaming terminology

In this section different terms used in presenting and solving lot streaming problems are explained.

**Constant, consistent and variable sublots**

Three types of sublots exist in lot streaming problems. When sizes of sublots between stages $i$ and $i$+1 are not equal to those between stages $i$+1 and $i$+2, sublots are variable (Trietsch and Baker, 1993). In other words, if we consider $q_{ij}$ as the size of sublot $j$ at stage $i$, in an $m$-stage production system with $k$ sublots, $q_{ij} \neq q_{i(j+1)}$, $i=1,\ldots,m$, $j=1,\ldots,k$-1 and $q_{ij} \neq q_{(i+1)j}$, $i=1,\ldots,m$-1, $j=1,\ldots,k$ concerning variable sublots (Chang *et al.*, 2004).

Consistent sublots have the same size in all the stages of their sequence which means: $q_{ij} = q_{j,i+1}$, $i=1,\ldots,m$-1, $j=1,\ldots,k$.

As the name states, size of constant sublots is the same in all the stages.

**Continuous and discrete version of sublots**

Different versions of sublots exist in various production environments. In some industries such as chemical industry, normally sublots are continuous and take real numbers; however, sublots can only be integer numbers of produced jobs in discrete part or product production systems.

**Intermittent idling and non-idling**

Machines can have idling while processing sublots of different jobs if intermittent idling is allowed. On the other hand, non-idling means that sublots should be processed consecutively on machines. Trietsch and Baker (1993) in an example show that intermittent idling corresponds to better solutions; however, in practice, this issue depends on different factors such as management point of view and problem definition.

**No-waiting and waiting**

No-waiting concerns the system in which jobs must be immediately transferred to the succeeding machine after their processing on the current machine. In wait schedules, sublots can wait between consecutive stages in a considered buffer (Feldmann and Biskup, 2006).

**Intermingling / non-intermingling**

When intermingling is allowed, sublots of one job can be mixed with those of other jobs. In other words, sublots of a particular job may be processed in the sublot sequence of another job. When non-intermingling exits, sequence of sublots of one product is started when processing of all sublots of the previous product are finished in that stage.

**Permutation flow shop**

In a permutation flow shop, sequence of jobs is the same in all the stages. In these problems, when the processing order of jobs is determined, it is followed throughout all the stages.

**Attached and detached setups**

In some production systems, setup of a job requires its arrival on a particular machine. In this case the setup is called attached. On the contrary, in detached setups we may setup a machine for a job in its idle time before the availability of the job.

**Sequence dependent / independent setups**

In a multiple-product flowshop setting, setups may be dependent on the change of one product to another. A machine needs to be equipped when the sublot of a new product is going to be processed. Therefore, setup time incurred for changing from job $i$ to job $j$ might be different from that of $i$ to $k$. Moreover, no machine setup is needed between two consecutive jobs of one family in a job shop environment (Potts and Van Wassenhove, 1992). It is called independent setups when setups are required between any two contiguous sublots on a particular machine.

### 1.6 Research in this thesis

Hybrid flow shop scheduling is one of the most common problems in manufacturing. This study contributes to scheduling and lot streaming of multiple products in a multi-stage hybrid flow shop in order to minimize the makespan.

#### 1.6.1 Scope and objectives of this thesis

The purpose of this research is to develop a mixed integer programming formulation to optimize the sequence of jobs and determine sublot sizes at the same time so that the makespan is minimized. Sublots of each job/product which can be continuous or discrete are considered to be consistent while intermingling is not allowed. The problem is studied under various settings and takes into account sequence independent setup times. The efficiency of the model is evaluated by carrying out several experiments and comparing the results with those of previous studies.

#### 1.6.2 Research contributions

This study presents a model that is an extension to the one of Biskup and Feldmann (2006) in lot streaming a single product in a multi-stage flow shop with variable sublots and their most recent work Feldmann and Biskup (2006); it deals with lot streaming of multiple jobs in flow shops. Zhang *et al.* (2005) develop two heuristic methods to solve the problem of streaming multiple jobs in a hybrid flow shop with two stages. Another heuristic method is presented in Liu (2008) for lot streaming of a single product in a two-stage hybrid flow shop. Both of the two studies consider identical parallel machines for the first stage and a single machine for the second stage. The proposed model in this study is the first mathematical model that incorporates lot streaming multiple products in a hybrid flow shop with multiple stages. The determination of sublot types based on the problem characteristics is studied. Moreover, the multiple-product problem is decomposed into single-product problems and the influences are analyzed. The main contribution of this thesis is to take into account any combination of parallel and single non-identical machines in hybrid flow shops which enables analyzing the corresponding makespan for selecting the best setting.

### 1.6.3 Organization of the thesis

Chapter 2 presents a review on the literature of lot streaming in different categories. In chapter 3, the proposed mathematical model is presented and discussions on different problem settings are given. Chapter 4 deals with extensive numerical examples and comparison with similar studies to illustrate the efficiency of the method. Lastly, concluding remarks and directions for future research are presented in chapter 5.

# Chapter Two Literature Review

## 2.1 Introduction

While traditional scheduling problems consider fixed lot sizes for products to transfer between stages in a production system, other approaches have been developed to accelerate the production flow. Reiter (1966) introduces lot streaming to allow overlapping of processes. This method increases efficiency of production systems, minimizes completion time, reduces lead time, and makes faster delivery to customers. Lot streaming related to manufacturing resources planning (MRP) and optimized production technology (OPT) is discussed in Lundrigan (1986). OPT combines MRP II and just-in-time systems and emphasizes on planning and controlling the resources. This technique has been studied and used in industry and shown to be effective for compressing manufacturing lead time. Kher *et al.* (2000) study the impacts of push and pull lot streaming approaches on material handling in stochastic flow shops. In implementing lot splitting, instead of push approach they apply pull approach which is more likely to be used in practice and show its benefits regarding inventory and customer service performance.

Lot streaming has been studied extensively in the last decade to best deal with real problems in industry. Bridging the gap of lot sizing and traditional scheduling problems, contribution of lot sizing and sequencing problems and flow-time reduction are important goals in industry that motivate researchers for implementing lot streaming problems (Bukchin *et al.* 2002). Various methods and approaches have been carried out to handle difficulties in lot streaming problems.

## 2.2 Problem structure and notation set definition

Existing lot streaming problems can be classified by their types. Potts and Van Wassenhove (1992) present a structure of three main dimensions and seven sub-dimensions. Sub-dimensions are categorized into several levels as well. Chang and Chiu (2004) introduce these classifications and notations that are used in the literature to denote different characteristics of lot streaming problems. Table 2-1 illustrates this classification.

**Table 2-1 Classifications of lot streaming problems**

| Main dimension | Sub-dimension | Level |
|---|---|---|
| System configuration($\alpha$) | Production type ($\alpha_1$) | Flow shop ($F_m$) |
| | | Job shop ($J_m$) |
| | | Open shop ($O_m$) |
| | Number of products ($\alpha_2$) | Single product ($L_1$) |
| | | Multiple-product ($L_n$) |
| Sublot-related feature ($\beta$) | Sublot type($\beta_1$) | Equal sublots ($E$) |
| | | Consistent sublots ($C$) |
| | | Variable sublots ($V$) |
| | Divisibility of sublot size ($\beta_2$) | Continuous version ($R$) |
| | | Discrete Version ($A$) |
| | Operation continuity($\beta_3$) | No idling case ($I_{no}$) |
| | | Idling case ($I$) |
| | Activities involved ($\beta_4$) | Setup ($S$) |
| | | Transportation ($M$) |
| Performance measurement ($\gamma$) | Performance criterion ($\gamma_1$) | For the time model:<br>Makespan ($C_{max}$)<br>Total flow time ($\sum \boldsymbol{F}$)<br>Mean tardiness ($\overline{\boldsymbol{T}}$)<br>Number of tardy jobs ($n_T$)<br>For the cost model:<br>Total cost ($TC$) |

The three field descriptor $\alpha|\beta|\gamma$ presented by Potts and Van Wassenhove (1992) includes $\alpha$, machine environment, $\beta$, job characteristics, and $\gamma$ the objective function. As depicted in Table 2-1, the seven sub-dimensions as $\{\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_1\}$. If an activity is excluded from the problem, "-" replaces its notation. If a certain sub-dimension comprises number of levels (e.g., no idling ($I_{no}$) and idling ($I$)), then $\beta_3 = I_{no} / I$.

As an example, $F_m$, $L_n$ / $V$ / $C_{max}$, represents a multiple-product lot streaming flow shop problem with variable sublots to minimize the total completion time.

## 2.3 Single-product problems

Trietsch and Baker (1993) introduce various models for solving different lot streaming problems. Continuous and discrete sublots, intermittent idling, consistent and variable sublots and capacitated transporter are discussed in their paper. They summarize the literature of the subject in each category and present related models and problem definitions; furthermore, they develop new linear programming formulations for two and three machines and generate new results. They claim that minimal makespan is achieved with variable sublots and intermittent idling.

Chen and Steiner (1997) study a discrete lot streaming problem in single-product multi-stage flow shops. Batches are assumed to be available at time zero and sublots are consistent. Network representation is used for the problem and the objective is to determine sublot sizes to minimize the length of the critical path of the network. Two approximation algorithms are developed and examples are presented to verify the tightness of the problem bounds. They demonstrate the effectiveness of their solution approach and solve the problem in $O(s)$ time, $s$ as number of sublots; it represents very good quality approximations for consistent sublots.

The two approximation procedures can be applied for the reverse type of the presented flow shop problem as well. They claim that the solution is very good in practice since it yields little machine idle time in the schedule.

Chen and Steiner (1998) study a single product lot streaming problem to minimize the makespan in multi-stage flow shops. Attached and detached setups are considered in their model. Batch availability is assumed, that is, a batch is transferred to the successive stage when the whole batch is finished processing in that stage. Based on the problem definition they develop a network representation to find sublot sizes that minimize the longest path, sum of weights of the vertices, in the network. A solution is given for a simple reduced auxiliary problem which considers detached setups. Further on, optimal sublot sizes are determined in a general problem of three machines. They introduce a complex problem considering detached setups solved in $O(s)$ time.

Chen and Steiner (1999) present a lot streaming problem to minimize makespan of a single job in a two-machine flow shop. They assume discrete sublots that are available at time zero. Considering no preemption between jobs, multiple-product problems are decomposed into single-product problems; therefore their single product model can be applied to a multiple-product lot streaming problem as well. They show that an optimal solution can be derived from the continuous problem and be solved in $O(s)$ time. Considering consistent sublots, a network representation is given and the objective is to minimize the longest path by determining the sublot sizes. They synchronize the schedules of the two machines which is desirable in practice in automated systems. Moreover, it is shown that increasing the number of sublots results in smaller makespan.

Sen and Benli (1999) study the problem of lot streaming with consistent sublots, they present models for single-job and multiple-job problems in open shops with two machines. For the single job problem they introduce two cases; *single routing* models where sublots of the product have the same routing (the order in which they are processed in machines) and *multiple routing* models when this routing varies. Where the routing is pre-defined, the problem turns into a flow shop lot streaming problem and the decision is to determine sublot sizes. On the other hand, when sublot sizes are given, the objective is to determine the routing of the job. They study and synthesize the models of the two types of routings for single and multiple products and evaluate their effects on reducing the makespan.

Sriskandarajah and Wagneur (1999) study a two-machine no-wait flowshop problem to minimize makespan with lot streaming. Single-product and multiple-product problems are studied and a heuristic method is developed to find a close-to-optimal solution. They develop a model for continuous sublots when the product lot is large and number of sublots is fixed. Based on this model they develop another model for a restrictive no-wait problem with discrete sublots. Furthermore, they extend their model to a more general lot streaming problem where number of sublots is also a decision variable. They solve a real world problem in an anodizing line, a flow shop of chemical processing tanks. They solve the problem for continuous size of sublots polynomially.

Kalir and Sarin (2001) consider the problem of lot streaming in a single batch flow shop. They discuss various limitations in applying lot streaming problems in reality and try to relax some of these constraints in their model. They put the number of sublots as a decision variable and incorporate both time and cost objectives in the problem. They also give a polynomial solution procedure for discrete sublots and consider the effects of attached setups on the makespan. The impact of transfer time on the time-based and cost-based objectives is also studied. Their solution approach enables the management to verify the number of sublots in a lot when machine setups are considered. Other objectives can also be adapted in their model such as minimizing mean flow time and work-in-process inventory as well as minimizing makespan in a cost model.

Bukchin *et al.* (2002) consider a single job lot streaming problem in a two-machine flow shop with detached setups. The objective is to minimize the total flow-time. The applications of their study with detached and attached setup times are mainly in semiconductor industry where setup is incurred when the whole sublot is available. To solve their non-linear model, they develop a procedure based on SMB, *single machine bottleneck*, an intuitive solution structure which can produce optimal or near optimal solutions.

Chen and Steiner (2003) discus the problem of single-job lot streaming in *m*-machine flow shop. They consider discrete consistent sublots with no-waiting between consecutive stages while their model can also have waiting constraints. An integer programming model is developed and minimizes the makespan. Since no polynomial-time solution is known for this solution, they use network representation to determine sublot sizes to minimize the longest path in the network. A solution for *m*-machine problem for two sublots is presented. They also present two polynomial-time approximation methods for the general case of discrete sublots. Computational results show that these methods are effective.

Chiu *et al.* (2004) consider a single product lot streaming problem in a multi stage production system. Transportation cost is incurred and objective is to determine batch sizes in order to minimize makespan and transportation cost. They discuss different lot streaming policies in the literature. They study the allocation of variable sublots and consider multiple transporters as well as simultaneous attached and detached setups. A general mathematical model is developed to determine the allocation of variable sublots. Since their binary mixed integer programming model is difficult to solve, they develop two heuristic methods. Results show that the heuristic methods are effective and efficient.

Chiu and Chang (2005) study a single-product multi-stage cost-related lot streaming problem in flow shops. They consider costs of raw material, WIP (work-in-process) and finished-product inventories to minimize the total annual cost by determining optimal batch sizes and propose two optimization models. The first model assumes costs of various inventory types as well as setup, transportation and finished product shipment costs. The cost of reducing the makespan is of great significance and is added up to build the second model which is more general. Two solution methods are presented to search for the optimal solution with an integer number of sublots. From the analysis of examples and computational results, inventory holing costs, setup costs and unit-processing-time show greater impact on the total cost than sublot transportation and shipment of finished products.

Biskup and Feldmann (2006) study the lot streaming problem of single product in an *m*-machine flow shop. Sublot availability is assumed and sublots can be constant, consistent or variable. They introduce their problem as MVS (multi-stage, variable sublots, sublot availability) and develop a mixed integer programming formulation. This is the first integer programming formulation for lot streaming with variable sublots. They also formulate attached and detached setups in the problem. The objective is to minimize makespan by determining sublot sizes for a given maximum number of sublots in all stages. In an example of five production stages, they show a better performance in applying variable sublots instead of constant and consistent sublots. This approach can lead to significant improvements especially when setups are incurred.

## 2.4 Multiple-product problems

Lee et al. (1997) considers lot streaming as one of the new trends in scheduling theory to extend classical algorithms to more closely related models to real problems. They indicate that in spite of difficulties in applying these results, they are motivated by industries. Some extensions and classifications of lot streaming with more practical constraints are presented as a part of deterministic scheduling in this paper. Various neighborhood search techniques and constraint-guided heuristic search techniques are reviewed as well. Simulated annealing, Tabu search and genetic algorithm are studied for NP-hard problems discussed in their paper.

Cetinkaya and Kayaligil (1992) study a multiple product two-machine lot streaming flow shop problem. Setups are sequence independent and separate from processing times. Transportation time between stages is negligible and the objective is to minimize the completion time. They construct a model similar to Johnsons' algorithm and generate an optimal solution.

Vickson and Alfredsson (1992) study the effect of lot splitting in a two and three machine multiple-job problem. Simple modifications are applied to Johnson's algorithm to develop a model for minimizing the makespan. Non-preemption schedule is assumed and jobs finish processing the whole sublot before transferring to the next stage. Transportation time and cost are not considered and setup times between jobs are negligible. They present empirical study to show the improvement of makespan in applying sublot transfer. The problem may be solved by branch and bound without backtracking, local neighborhood search or LIT heuristic. Examples are given to show the better performance of job splitting.

Cetinkaya (1994) consider the lot streaming problem in two-stage flow shops for multiple products. Job sizes are not necessarily equal but have a maximum number of allowable transfer batches. Batches are available at time zero and are transferred from the first to the second stage. Preemption is not allowed. They integrate sublot size decisions and scheduling decisions when setups are independent of job sequence and the objective is to minimize the maximum flow time (makespan). In order to do that, they decompose the problem to a sequencing problem and a batch sizing problem to find optimal solutions for both sub-problems. Afterwards, an algorithm is proposed for the combined problem which is solvable in polynomial time.

Baker (1995) studies a multiple-job lot streaming problem in a two-machine flow shop. All jobs consist of identical items processed in machines without preemption. Sublots are equal in size while unit-size sublots are also considered. Setup settings comprise both types of attached and detached setups and time lags can be applied to the problem as well. The objective is to minimize the makespan and gives the best response for transfer batches of unit size. They review and synthesis the existing flow shop solutions to develop an approach for getting the optimal solution for the considered lot streaming problem.

Vickson (1995) study a multiple product lot streaming problem in a two-machine flow shop. Setups are assumed to be sequence independent which can be attached as well as detached. Transfer times of sublots are finite and sequence independent. He considers limited material handling capacity. In case of no preemption and when the objective is makespan minimization, the sequencing problem for two machines is solvable by Johnson's algorithm. They develop closed form solutions for continuous sublots considering setups. A linear integer programming model is presented for discrete sublots which is solvable in polynomial time. The procedure for the two types of setups is almost the same.

Sriskandarajah and Wagneur (1999) consider the lot streaming problem with multiple products in two-machine no-wait flow shops. Sublots are both assumed to be integer and continuous. The objective is to determine sublot sizes and sequence the jobs simultaneously. They develop a heuristic method for optimal scheduling and lot sizing multiple products and results are shown to be close to optimal. The model is extended to solving problems where the number of sublots is also a decision variable.

Subodha *et al.* (2000) consider lot streaming problems for multiple products in *m*-machine flowshops. The objective is to minimize the makespan. They present a linear programming model for the single product problem assuming fixed number of continuous sublots. For integer size of sublots an existing heuristic is used to determine sublot sizes. As they move on to the multiple product and continuous-sized sublots, a TSP (Traveling salesman problem) formulation is presented as one of the approaches for obtaining local optimal solutions. Another heuristic algorithm is proposed for the integer size of the sublots. For the interacting decision of lot streaming and sequencing multiple products, a genetic algorithm is developed which is shown not to yield a strong solution; however, this meta-heuristic is also used to optimize the number of sublots.

Kalir and Sarin (2001) consider lot streaming problems for multiple products in multi-stage flow shops. Sublots are considered to be equal and intermingling of sublots is not allowed. The objective is to minimize the makespan by optimizing the job sequence. A heuristic method is developed to achieve a near-optimal solution for this problem. The "bottleneck minimal idleness" heuristic maximizes the time buffer before the bottleneck and at the same time sequences the larger sublots earlier in the sequence. An example problem in surface mount technology (SMT) is given to illustrate the impact of sequencing in lot streaming problems. They show the efficiency of the model in an experimental study and compare it to another heuristic method named FIH (fast insertion heuristic).

Yoon and Ventura (2002) study a lot streaming problem in flow shops to minimize the mean absolute deviation from due dates for multiple products. Sublots can be either constant or consistent. Buffer capacities between successive stages can be limited or infinite. A linear programming formulation for infinite buffer capacity is developed to find optimal completion times of sublot. The model is then extended to finite capacities and no-wait flow shop. Afterwards, they apply several pairwise interchange methods to find near-optimal solutions. They use computational experiments to illustrate the effect of different types of buffers and sublots in solving the lot streaming problem. Consistent sublots and infinite buffers are shown to yield better results.

Yoon and Ventura (2002) apply genetic algorithms to multiple-product multi-stage lot streaming problems. Sublots are assumed equal in size and buffer capacity is infinite. The objective is to minimize the mean weighted absolute deviation from due date. Given an initial job sequence to the developed linear programming formulation, start and completion times of sublots are obtained. Since genetic algorithm has some weaknesses such as premature convergence, a hybrid genetic algorithm (HGA) is used to search among different sequences and find the best solution. This meta-heuristic applies a Non-Adjacent Pairwise Interchange (NAPI) method as well as the LP formulation to obtain the optimal solution. Lastly, the efficiency of the method is compared with similar heuristic methods. The proposed HGA is shown to perform well for this type of problems.

Feldmann and Biskup (2006) study multiple-product multi-stage lot streaming in a permutation flow shop. Sublots are assumed consistent and can intermingle. The objective is to minimize makespan and a mixed integer programming formulation is proposed. The model determines optimal sublot sizes and optimal job sequence at the same time. They compare the results of lot streaming problem when intermingling is allowed versus non-intermingling. It is shown to be beneficial allowing sublots of different products to intermingle. This integer programming model is capable of solving medium size problems.

## 2.5 Hybrid flow shop scheduling problems

Gupta and Tunc (1991) study the problem of scheduling multiple jobs in a two-stage hybrid flow shop. The first stage is assumed to have one machine while the second stage has parallel identical machines. The objective is to determine the sequence of jobs in order to minimize the makespan. They solve the problem for two cases: 1) the number of jobs is equal to or less than the number of machines in the second stage; 2) the number of jobs exceeds the number of machines in the second stage. For the first case, the longest processing time (LPT) scheduling rule is applied which yields an optimal solution. Two heuristic algorithms are presented to minimize the total throughput time of all jobs in two stages. These approximate algorithms are polynomially bounded. Computational results indicate that the efficiency of the proposed algorithms increases by the increment of number of jobs and can be used in solving large-size problems.

Yalaoui and Chu (2003) study a parallel machine scheduling problem with multiple jobs when lot splitting is allowed. Setups are sequence-dependent and objective is to minimize the makespan. They develop a heuristic method of two parts. First, a reduced single machine problem is presented and transformed to a Travelling Salesman Problem (TSP). Second, the results of the TSP problem are used as an initial solution which is improved step by step.

Logendran and Subur (2004) study the scheduling problem of multiple jobs in unrelated parallel machines when job splitting is applied to minimize total weighted tardiness. Sublots of a batch have the same release time, weight and due date. Preemption is allowed and machine idleness without any cost is permitted as well. A mixed integer linear programming model is developed to incorporate constraints for tight due dates, high priority and high workload of products in a just-in-time manufacturing system. They present a solution algorithm that first identifies the initial solution and then uses Tabu search to determine the near-optimal /optimal solution. A variable Tabu list for small size problems and a fixed Tabu list for large size problems are proposed in their paper. They show a good quality of results for their method with short computational time.

Zhang *et al.* (2005) study multiple-product lot streaming problems in two-stage hybrid flow shops. The first stage has $m$ identical machines while the second stage has a single machine. Sublots are consistent and intermingling is not allowed. The objective is to minimize mean completion time through determining the number of sublots of the jobs, their sizes and the sequence. Two heuristics are proposed with the same strategy that first sequence the jobs and then split and schedule them. To determine the lower bound of the solution a mixed integer linear programming model is formulated. Further on, they evaluate the performance of the methods by conducting extensive experiments.

Liu (2008) study the lot streaming problem of a single product in a two stage hybrid flowshop. First stage has $m$ identical machines while the second stage has one machine. Sublots are considered to be equal and processed in both stages. Optimizing sublot sizes is done by a linear programming formulation. It is transformed to a convex optimization problem which is solved by two heuristics. Their heuristics can provide close to optimal solutions in a wide range of experiments.

Ruiz and Maroto (2006) consider a complex scheduling problem that could be a generalization of other problems such as permutation flowshop or flowshop with multiple processors (FSMP). They propose a genetic algorithm for a hybrid flowshop with unrelated parallel machines. Setups are sequence dependent and machine eligibility is considered, that is, not all products may be processed by all the machines at a given stage. The genetic algorithm presents new characteristics and new crossover operators. To evaluate the efficiency of the method they carry out extensive experiments and perform adaptations of some other meta-heuristics that show better performance in similar production environments. Since the studied problem contributes to a common problem in textile and ceramic tiles production, they also conduct experiments given real data to their model and demonstrate improvement in the makespan of schedules.

## 2.6 Concluding Remarks

Lot streaming has been studied for decades as a technique in lot sizing and scheduling problems to improve the performance of production lines. Various approaches have been carried out to overcome the difficulties in solving lot streaming problems and make them capable of solving large-size problems.

There has been much research work for single-product lot streaming problems, assuming several assumptions and objectives and presenting linear programming formulations as well as heuristic methods. Few recent studies have been conducted to present integer programming models for multiple-product problems and more general heuristic methods are yet to be developed for large-size problems. Few research works consider lot streaming in parallel machine or hybrid flow shops. Some researchers study the two-machine hybrid flow shop problems and methods need to be discovered to solve general lot streaming problems in hybrid or flexible flow shops.

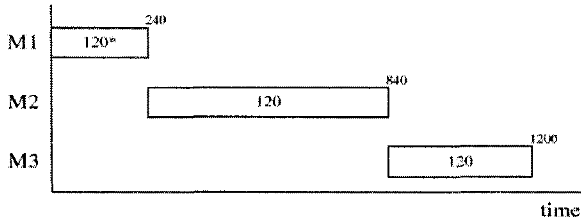
In the following chapter an integer programming model is developed for multiple-product lot streaming problem in hybrid flow shops. Products may skip some stages. Optimal sublot sizes as well as an optimal sequence of sublots and products are determined to minimize the makespan.

# Chapter Three Model Formulation and Solution Approach

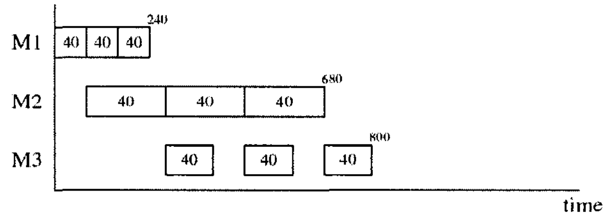
## 3.1 Problem introduction

Assume that in a manufacturing flow shop $J$ jobs/products are processed in $S$ stages. Jobs consist of several identical items which have the same processing time on each stage. All jobs have the same sequence of processing and are processed by the order of stages (i.e. 1, 2, …, $S$). Lot streaming is applied to accelerate the flow of the line and to decrease the makespan.

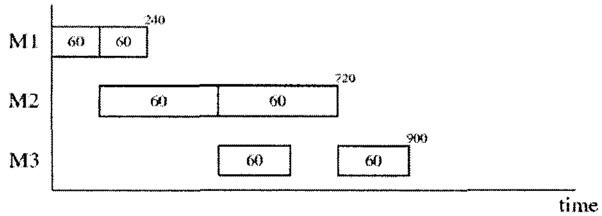
Chang *et al.* (2004) illustrate the advantages of lot streaming using an example. There are three stages in a flow shop. Processing times on stages 1, 2 and 3 are 2, 3, and 5 minutes, respectively. A batch of 120 units is processed on these stages and setup cost is negligible. Five scenarios are considered: without splitting, splitting into 2 constant sublots, splitting into 3 constant sublots, consistent sublots, and variable sublots named as schedules 1, 2, 3, 4 and 5, respectively. As depicted in Figure 3-1, makespan of the first schedule is 1200 time unit while it is reduced to 900 and 800 by applying constant sublots. In this example for schedules 4 and 5 which are consistent sublots and variable sublots, respectively, the completion time is slightly increased; however, generally these scenarios have better responses in lot streaming problems. In this example we see the improvement of 33.33% in makespan of schedule 3 by applying lot steaming technique.



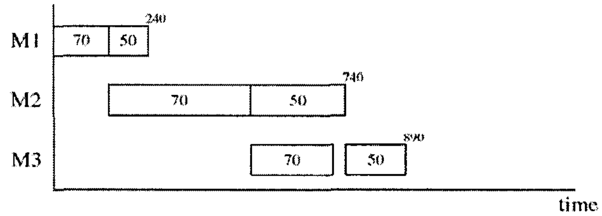

Schedule 1- Without splitting



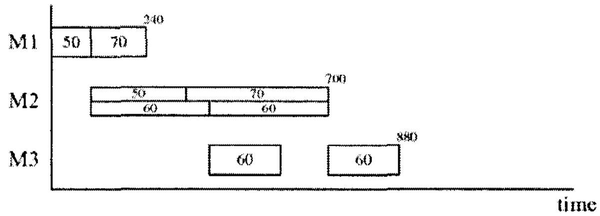

Schedule 3- Splitting into 3 constant sublots




Schedule 2- Splitting into 2 constant sublots




Schedule 4- Consistent sublots




Schedule 5- Variable sublots

**Figure 3-1 An example of lot streaming problem with different types of sublots**

Based on the problem characteristics and implementation concerns, different types of sublots are considered in lot streaming problems. Variable sublots might be efficient where setups are incurred for processing sublots on machines. The advantage of applying variable sublots is that sublots may be split less on machines with high setup time so that the completion time is decreased. On the other hand, there are several disadvantages that discourage researchers in using variable sublots such as increased computational time with insignificant solution improvement and difficulties of implementation in solving real world problems.

In hybrid flow shops stages have either single or parallel processing machines. Parallel machines are used to overlap processes in stages with low production flow. Sublots which are not allowed to be overlapped on a single machine in lot streaming problems might be processed simultaneously on parallel machines. In other words the constraint to prevent overlapping of sublots is relaxed on parallel machines and the chance of reducing the makespan is increased.

Researchers consider various methods to assign sublots to hybrid stages and to sequence their processing. A method called "rotation" is used by Liu (2008) to sequence the sublots of a product in a two stage hybrid flow shop. In the considered problem, the first stage has parallel identical machines whereas the second stage has a single machine. In this method, sublots are assigned to parallel machines based on the order of machines; sublot 1 to machine 1, sublot 2 to machine 2,..., sublot $i$ to machine $m$, sublot $i$+1 to machine 1, and so on. These sublots are transferred to stage 2 and are processed on the single machine in the same order. Rotation method is shown to find the optimal solution in this type of hybrid flow shop problem. Figure 3-2 illustrates this sublot assignment for eight sublots.

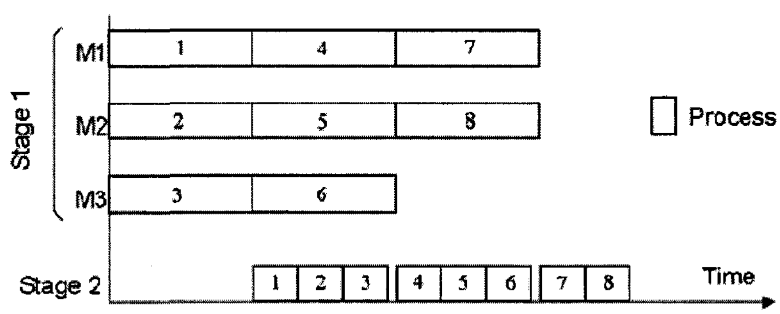



**Figure 3-2 Sublots assignement by rotation method**

## 3.2 Mathematical model

### 3.2.1 Problem description

As discussed in previous chapters, we intend to increase the scope of studies in hybrid flow shop lot streaming problems. Consider multiple products that are processed in a multi stage hybrid flow shop. Machines are non-identical and number of them in parallel stages is determined. The problem is to decide sublot sizes of products and sequence them on machines so that the completion time of the last sublot in the last stage is minimized. As such, optimum number and size of sublots as well as optimum sequence of products are to be decided in our model.

### 3.2.2 Model assumptions

In lot streaming problems, researchers consider several assumptions based on their problem descriptions. We consider following assumptions in this study:

1. Discrete or continuous sublots are consistent and available at time zero.
2. Sublots of a particular product are assigned to machines and are processed where a machine in that stage is eligible and ready for setup and processing.
3. Each sublot is processed on only one machine at a time in parallel stages.
4. Each product may have different number of sublots.
5. Intermingling is not allowed, that is sublots of different jobs are not mixed while being processed on machines.
6. Permutation flow shop is assumed and jobs have the same processing order in all the stages.

7. Each machine can process at most one job at a time and each job cannot be processed on more than one machine at a time.
8. Setups are attached and sequence independent.

### 3.2.3 Model notations

Indices:

$j, k$ Indices for products, $j, k = 1, \ldots, J$, where $J$ is number of products.

$i, t, r$ Indices for sublots, $i, t, r = 1, \ldots, SP_j$, where $SP_j$ is maximum number of sublots for product $j$.

$s$ Indices for stages, $s = 1, \ldots, S$, where $S$ is number of stages.

$m, f$ Indices for machines in each stage, $m, f = 1, \ldots, M$, where $M$ is maximum number of machines in all the stages.

Parameters:

$U_j$ Number of identical items of product $j$

$SM_s$ Number of machines available in stage $s$

$P_{jsm}$ Processing time for one unit of product $j$ in stage $s$ on machine $m$

$ST_{jsm}$ Setup time for product $j$ in stage $s$ on machine $m$

$G$ Sufficiently large number

Variables:

$C_{jism}$ Completion time of $i^{th}$ sublot of product $j$ in stage $s$ on machine $m$

$L_{jism}$ Sublot size of $i^{th}$ sublot of product $j$ produced in stage $s$ on machine $m$

$B_{jism}$ $\begin{cases} 1, & if \text{ sublot } i \text{ of product } j \text{ is produced in stage } s \text{ on machine } m \\ 0, & otherwise \end{cases}$

$X_{jikt}$ $\begin{cases} 1, & \text{if sublot } i \text{ of product } j \text{ is sequenced prior to sublot } t \text{ of product } k \\ 0, & otherwise \end{cases}$

$Z$ Maximum completion time of all the products

### 3.2.4 Model

In this study the model is formulated as:

Minimize $Z$ subject to

$$Z \geq C_{jiSm}\,;\ j = 1, \dots, J, i = 1, \dots, SP_j, m = 1, \dots, M \tag{3-1}$$

$$\sum_{i}^{SP_j} \sum_{m}^{M} L_{jism} = U_j;\ j = 1, \dots, J, s = 1, \dots, S \tag{3-2}$$

$$L_{jism} = 0;\ \ j = 1, \dots, J,\ \ i = 1, \dots, SP_j,\ \ s = 1, \dots, S,\ \ m > SM_s \tag{3-3}$$

$$C_{jism} \geq P_{jsm}L_{jism} + ST_{jsm}B_{jism};\ \ j = 1, \dots, J, i = 1, \dots, SP_j,\ \ s = 1, \dots, S,\ \ m = 1, \dots, M \tag{3-4}$$

$$C_{jism} - P_{jsm}L_{jism} - ST_{jsm}B_{jism} \geq C_{jrsm};\ \ j = 1, \dots, J, i, r = 1, \dots, SP_j,\ \ r < i, s = 1, \dots, S,\ \ m = 1, \dots, M \tag{3-5}$$

$$C_{k1sm} - P_{ksm}L_{k1sm} - ST_{ksm}B_{k1sm} \geq C_{jism} - \left(1 - X_{jikt}\right)G;\ \ j, k = 1, \dots, J, j < k, i = 1, \dots, SP_j, t = 1, \dots, SP_k, s = 1, \dots, S,\ \ m = 1, \dots, M \tag{3-6}$$

$$C_{j1sm} - P_{jsm}L_{j1sm} - ST_{jsm}B_{jism} \geq C_{ktsm} - X_{jikt}G;\ \ j, k = 1, \dots, J, j < k, i = 1, \dots, SP_j, t = 1, \dots, SP_k,\ \ s = 1, \dots, S,\ \ m = 1, \dots, M \tag{3-7}$$

$$C_{jism} - P_{jsm}L_{jism} - ST_{jsm}B_{jism} \geq C_{jis-1,f};\ \ j = 1, \dots, J, i = 1, \dots, SP_j, s = 2, \dots, S, m, f = 1, \dots, M \tag{3-8}$$

$$X_{jikt} \leq X_{ji-1,kt-1};\ \ j, k = 1, \dots, J, j < k, i = 2, \dots, SP_j, t = 2, \dots, SP_k \tag{3-9}$$

$$\sum_{m}^{M} B_{jism} \leq 1;\ j = 1, \dots, J, i = 1, \dots, SP_j, s = 1, \dots, S \tag{3-10}$$

$$L_{jism} \leq B_{jism}G;\ j = 1, \dots, J, i = 1, \dots, SP_j,\ \ s = 1, \dots, S,\ \ m = 1, \dots, M \tag{3-11}$$

$$\sum_{m}^{M} L_{jism} \leq \sum_{m}^{M} L_{jis-1,m}\,;\ \ j = 1, \dots, J, i = 1, \dots, SP_j, s = 2, \dots, S \tag{3-12}$$

$$B_{jism} \in \{0,1\};\ \ j = 1, \dots, J, i = 1, \dots, SP_j, s = 1, \dots, S,\ \ m = 1, \dots, M \tag{3-13}$$

$$X_{jikt} \in \{0,1\};\ \ j,k = 1,\dots,J, j < k,\ \ i = 1,\dots,SP_j, t = 1,\dots,SP_k \tag{3-14}$$

Constraint (3-1) defines the completion time of the last sublot in the last stage $S$. This completion time is minimized in the objective function. Constraint (3-2) ensures that the total number of items produced in each stage for product $j$ should be equal to $U_j$. Constraint (3-3) sets the sublot size $L_{jism}$ to 0 if a machine does not exit in a stage. In our mathematical model $M$ is taken as the maximum number of machines available in all the stages; on the other hand, attribute $SM_s$ defines the exact number of machines in stage $s$ so when machine index $m$ exceeds $SM_s$ the corresponding lot size is set to 0 to avoid infeasible sublots. Completion time of each sublot should be at least greater than its processing time plus the required setup time. When a sublot is not assigned to a machine the binary variable $B_{jism}$ is 0 to prevent the equation from adding the corresponding setup time; this is defined by constraint (3-4). With (3-5) overlapping sublots of products on machines is prevented whereas sublots may be processed simultaneously on parallel machines. As such, sublot $i$ on machine $m$ should be processed after its required setup time and completion of any predecessor sublot $r$. We assume $r$ less than $i$ to avoid occurrence of cross-precedence and consider only possible alternatives. Constraints (3-6) and (3-7) determine the sequence of sublots of products. (3-6) is binding as long as $X_{jikt}$ takes the value 1. In a permutation flow shop, no index of stages or machines is needed for this variable since the product sequence is determined only once, regardless of stages and machines. Sublots are not intermingled and when product $j$ precedes product $k$ on a particular machine, first sublot of product $k$ is started after its required setup time and completion of sublots of product $j$. (3-6) and (3-7) are complementary constraints. Constraint (3-8) ensures that sublots of the same product do not overlap on any of the machines in consecutive stages; in other words, sublot $i$ of product $j$ is started in a particular stage when the required setup is completed and its process on any of the machines of the predecessor stage is finished. Constraint (3-9) controls binary variable $X_{jikt}$; if a sublot of product $j$ is not processed prior to that of product $k$, the successor sublot of each product should not be processed either. By controlling binary variable $B_{jism}$ in constraint (3-10) we ensure that a sublot is processed on at most one machine in each stage; it is not efficient, from management point of view, to setup parallel machines for processing a sublot of a certain product in a stage. Moreover, the optimal solution may consider less number of sublots for a product than $SP_j$ and set the binary variable $B_{jism}$ to 0 for sublot $i$ in stage $s$. Constraint (3-11) relates sublot sizes to $B_{jism}$ and sets them to 0 when a sublot is not processed on a particular machine. Constraint (3-12) is necessary for consistency of sublots and keeps the size of sublot $i$ of products consistent in consecutive stages.

To the best of our knowledge, the problem under study is most probably NP-hard (Trietsch and Baker, 1993; Biskup and Feldmann, 2006). Computational time increases by the number of sublots, the number of machines and stages, and the number of products. Sublots are usually considered continuous in lot streaming literature, however, discrete sublots can be generated by non-negative integer values for $L_{jism}$, $j= 1,\ldots, J$, $i= 1,\ldots, SP_j$, $s= 1,\ldots, S$, $m= 1,\ldots, M$. In this model, non-idling can be dealt with by changing the inequalities to equalities in constraint (3-5). In this case, a sublot starts its process on a certain machine right after the predecessor sublot is completed.

### 3.3 Model variations

The proposed model can be varied by changing some constraints and/or defining new variables so that other goals or problem assumptions are covered. The model can also be relaxed slightly to speed up and ease the calculation.

#### 3.3.1 Minimizing the mean completion time

As previously introduced, several objective functions are considered in scheduling problems. Mean completion time has also been studied by researchers in multiple product lot streaming problems. In our proposed formulation we can use the variable $Z_j$ to determine the completion time of each product and substitute the first constraint for:

$$Z_j \geq C_{jiSm}, j = 1, \dots, J, i = 1, \dots, SP_j, m = 1, \dots, M \tag{3-15}$$

The average of $Z_j$ is minimized in the objective function of the mean completion time as follow:

$$Min \sum_j^J \frac{Z_j}{J} \tag{3-16}$$

#### 3.3.2 Variable sublots in single product hybrid flow shop lot streaming problem

Another variation of the model enables us to consider variable sublots. This setting is advantageous where high setup time is incurred for processing parts on machines. By changing some constraints and adding new constraints concerning a new binary variable we develop a new formulation for the lot streaming problem of single product in multi-stage hybrid flow shops. This model is an extension to the integer programming formulation in Biskup and Feldmann (2006) for single product lot streaming problem in flow shops.

We take into account all the assumptions of the model in Section 3.2, and introduce the new binary variable as below:

$Y_{ist}$ $\begin{cases} 1, \ if \text{ sublot } i \text{ in stage } s \text{ is started after sublot } t \text{ in stage } s-1 \text{ is finished} \\ 0, \ otherwise \end{cases}$

Product index $j$ is removed from variables in all the constraints. As such, for instance $C_{ism}$ is the completion time of processing sublot $i$ in stage $s$ on machine $m$. Constraints (3-1) to (3-5) hold for the single product model while constraints (3-6) and (3-7) for determining sequence of products are removed. Constraint (3-8) substitutes for:

$$C_{ism} - P_{sm}L_{ism} - ST_{sm}B_{ism} \geq C_{ts-1,f} - (1 - Y_{ist})G, \ \ i,t = 1, \dots, I, s = 2, \dots, S,$$
$$m, f = 1, \dots, M \qquad (3\text{-}17)$$

To prevent overlapping sublots in consecutive stages sublot $i$ in the current stage should be started after completion of sublot $t$ in the preceding stage on any machine $f$. Therefore, this constraint is binding when $Y_{ist}$ is equal to one. (3-9) is removed, (3-10) and (3-11) hold for controlling binary variable $B_{jism}$. Following constraints are necessary to control binary variable $Y$.

$$Y_{is1} = 1, i = 1, \dots, I, s = 2, \dots, S \qquad (3\text{-}18)$$

$$Y_{ist+1} \leq Y_{ist}, i = 1, \dots, I, t = 1, \dots, I-1, s = 2, \dots, S \qquad (3\text{-}19)$$

Any sublot in stage $s$+1 should start after at least the completion of the first sublot in stage $s$. Furthermore, if sublot $t$ in stage $s$ is not started after completion of $i^{th}$ sublot in stage $s$-1, the consecutive sublot, $t$+1, should not be started either.

To determine the size of a particular sublot we cannot relate it simply to the size of other sublots in other stages as we do for constant or consistent sublots. For this purpose, another constraint should be concerned. If sublot $i$ in stage $s$ is started after sublot $t$ in stage $s$-1, its size should not exceed the sum of all sublots that are processed before the $t^{th}$

sublot in stage $s$-1 and sum of sublots produced in the current stage prior to $i^{th}$ sublot. This can be defined by the following constraint:

$$L_{ism} \leq \sum_u^t \sum_m^M L_{us-1,m} - \sum_u^{i-1} \sum_m^M L_{usm} + (1 - Y_{ist} + Y_{ist+1})G,\ i = 1, \dots, I, t = 1, \dots, I-1, s = 2, \dots, S \quad (3\text{-}20)$$

Regarding the setup time on a certain machine, this model determines the number and the size of sublots in each stage; to minimize the makespan, the whole batch may not be split due to the high setup time on a particular machine. Although the single product problem is less complex, it can be N-P hard when the number of stages, machines and sublots is increased.

### 3.3.3 Contributions to the developed models

In developing a mathematical model, where possible, we may relax some constraints to speed up calculations. However, relaxation must be performed in a way so that the applicability of the model is not destroyed. In the presented model in Section 3.2 we may relax constraint (3-10) and consider only producing fixed number of sublots per product and substitute it for:

$$\sum_m^M B_{jism} = 1;\ j = 1, \dots, J, i = 1, \dots, SP_j, s = 1, \dots, S \quad (3\text{-}21)$$

Determining the exact number of sublots instead of maximum number of them per product, a sublot is only built on one machine in a particular sage and is transferred to the consecutive stage. The advantage of this relaxation is reducing the computational time of the model while the decision on the optimal number of sublots is not made by the model any more.

Developed models in Sections 3.2 and 3.3.2 do not consider the rotation method; on the other hand, they assign sublots to machines based on the eligibility of machines rather than the order of machines.

# Chapter Four Numerical Examples and Analysis

In this chapter we present several examples to illustrate and validate the mathematical models presented in the previous chapter. We also analyze the proposed models under various problem settings based on sample input data. The data used in the examples are hypothetical and some of them are based on those from the literature. Several comparisons are performed to the existing problems in the literature of lot streaming. The model is programmed and solved by LINGO optimization software, version 8, on Compaq Pentium 4, CPU 2.93GHz.

## 4.1 Lot streaming single product in two-stage hybrid flow shop

An example problem is presented in Liu (2008) to verify the heuristic method introduced in that paper. The hybrid flow shop has two stages; two parallel identical machines in the first stage and a single machine in the second stage. A batch of 4000 identical items of a product is processed in the two stages. Processing time in the first stage is 1 and setup time is negligible while in the second stage, processing time is negligible and setup time is 2000. Sublots are considered constant as well as consistent and are greater than 1000 units. Rotation method is used to sequence sublots in this hybrid flow shop.

Figure 4-1 shows the Gant charts for global optimum solutions when constant and consistent sublots are used. Sublot sizes of 2000 are built in the problem with constant sublots. Assuming consistent sublots, sublot sizes are 1000 and 3000, respectively, for the first and second sublots. Improvement in makespan, $C_{max}$, from 6000 to 5000 is achieved by applying consistent sublots instead of constant sublots.

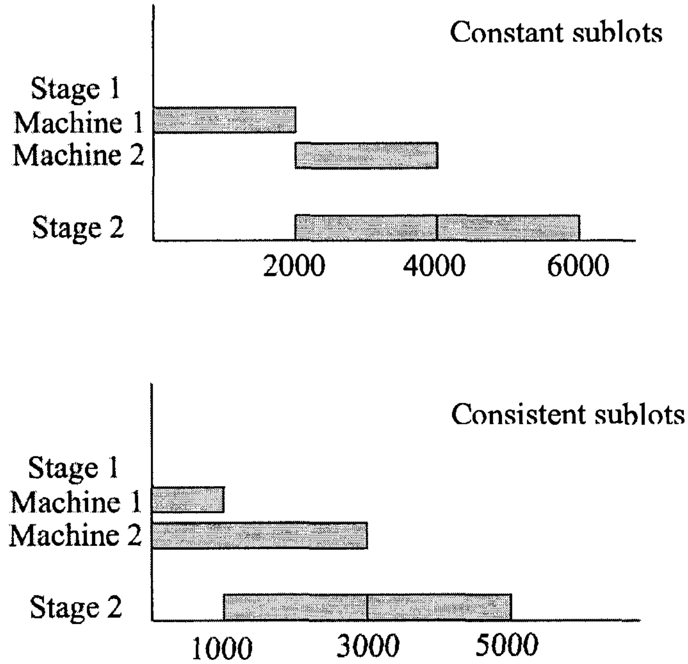



**Figure 4-1 Constant and consistent Sublots**

### 4.1.1 Impact of variable sublots

We intend to study the effect of applying variable sublots in this example. The data is input into our proposed model in Section 3.3.2 and the model is run without taking into account the lower bound of 1000 for sublot sizes. Global optimal solution in Figure 4-2 depicts that the two sublots of 2000 processed simultaneously in the first stage are combined together in one lot of 4000 due to the high setup time in the second stage. As a result, the makespan is decreased to 4000. It shows that the proposed model performs better and advantage of 33.3% in solution is achieved by applying variable sublots versus constant sublots.

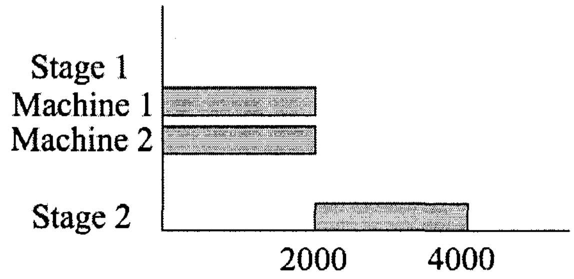



**Figure 4-2 Decrease in makespan by applying variable sublots**

### 4.1.2 Optimum number of sublots by using different types of sublots

Increasing the number of sublots may have various effects on the objective function value concerning different types of sublots in the example under study. In this Section we increase the number of sublots, holding other parameter values constant, and run the proposed model in Section 3.3.2. Experiments are implemented considering 3, 4, 5, and 6 as the maximum number of sublots for constant, consistent and variable sublots. Global optimum objective function values with their corresponding advantage over the original problem without lot streaming are illustrated in Table 4-1.

**Table 4-1 Comparison of constant, consistent, and variable sublots**

| Sublots | Constant | | Consistent | | Variable | |
|---|---|---|---|---|---|---|
| | Objective | Advantage % | Objective | Advantage % | Objective | Advantage % |
| 3 | 7333.33 | -22.22% | 5000 | 16.67% | 4000 | 33.33% |
| 4 | 9000 | -50.00% | 5000 | 16.67% | 4000 | 33.33% |
| 5 | 10800 | -80.00% | 5000 | 16.67% | 4000 | 33.33% |
| 6 | 12666.67 | -111.11% | 5000 | 16.67% | 4000 | 33.33% |

The advantage percentage is calculated as the makespan difference of original problem and each alternative over makespan of the original problem, multiplied by 100. For consistent and variable sublots the optimal solution will not change; sublot of 1000 and 3000 will be built for the first and second consistent sublots, respectively, while two sublots of 2000 in the first stage combine together in one lot of 4000 in the second stage using variable sublots. Conversely, if we limit our model to constant sublots, the objective function value increases drastically when the number of sublots is increased and the corresponding advantage percentage is negative. In general, the makespan is decreased by increasing number of sublots, although it is opposite in this problem because of the high setup time in the second stage. Therefore, it is optimal to produce the whole lot splitting into only two sublots. The trend of the advantage is depicted in Figure 4-3.

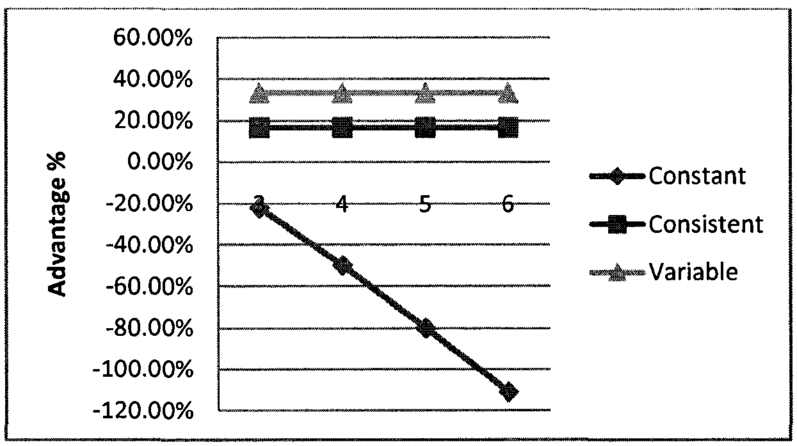



**Figure 4-3 Comparing advantages of using different types of sublots**

### 4.1.3 Impact of using consistent and variable sublots

To further study the impact of using different types of sublots in this type of hybrid flow shop lot streaming problem processing a single product, we introduce another example and analyze the results. A production line consists of two hybrid stages with 3 parallel identical machines in the first and a single machine in the second stage. A product of 6000 units is processed in the two stages. Processing times for each unit of the product are 3 and 1 in the first and second stages respectively. Setup times of sublots on machines are 3 and 5, respectively, in stages 1 and 2. Sublots are considered constant, consistent, or variable and maximum number of sublots is 3.

Optimal objective function values are: 12018 using constant sublots, 10924.36 for consistent sublots and 10919.36 for variable sublots. Figure 4-4 shows the schedules for consistent and variable sublots in this example. Consistent sublots of 1635.5, 2182.3, and 2182.3 are produced as the first, second and the third sublots respectively. The second and third sublots in stage 1 are combined together in one sublot of 4364.5 when variable sublots are considered. It is shown that applying variable sublots in hybrid flow shop lot streaming problems improves the optimal solution by varying number and size of sublots in different stages.

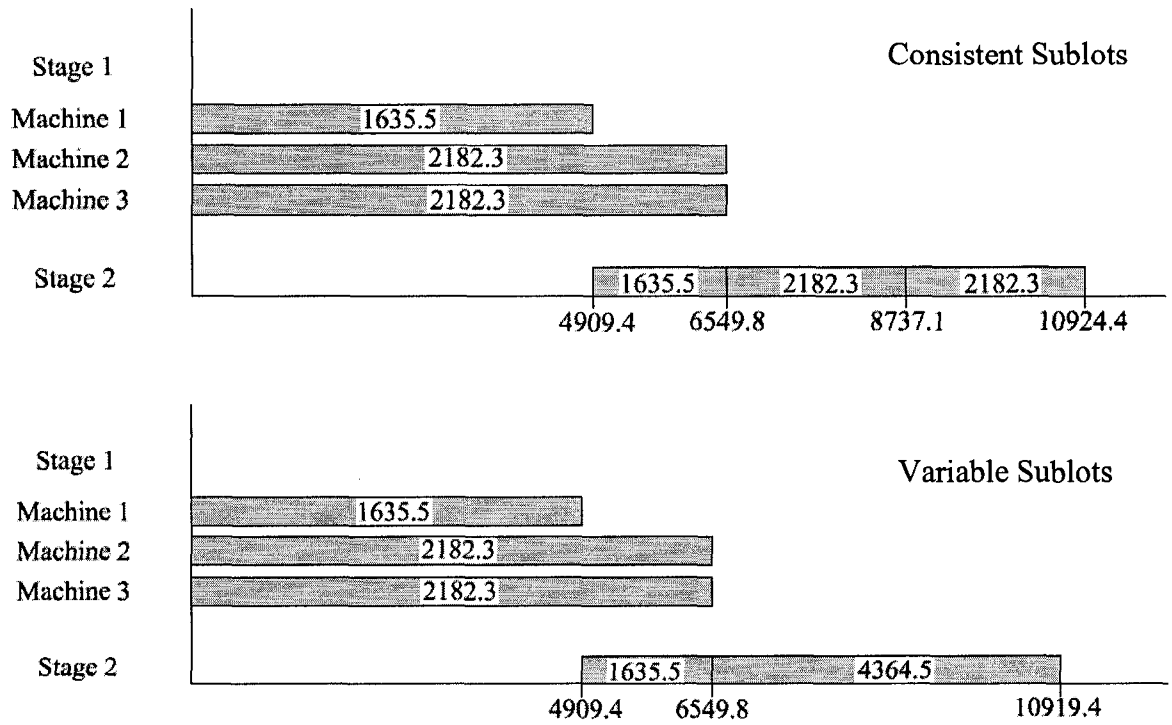



**Figure 4-4 Gant charts considering consistent sublots and variable sublots**

Selecting types of sublots in lot streaming problems is related to problem characteristics; as studied in a prior example, setup time on a particular machine influences sublot size determination. If we change the setup time on the single machine of stage 2 in the example, different makespan values can be generated by decreasing the setup time from 5 to 0 using consistent and variable sublots. Advantage over the problem without lot streaming is calculated and results are shown in Table 4-2.

**Table 4-2 Comparing the results by decreasing the setup time**

| Setup Time | Consistent | | Variable | | Advantage |
|---|---|---|---|---|---|
| | Makespan | Advantage % | Makespan | Advantage % | Δ % |
| 5 | 10924.36 | 54.50% | 10919.36 | 54.52% | 0.021% |
| 4 | 10921.91 | 54.51% | 10917.91 | 54.52% | 0.017% |
| 3 | 10919.45 | 54.51% | 10916.45 | 54.53% | 0.012% |
| 2 | 10917 | 54.52% | 10915 | 54.53% | 0.008% |
| 1 | 10914.55 | 54.53% | 10913.55 | 54.53% | 0.004% |
| 0 | 10912.09 | 54.54% | 10912.09 | 54.54% | 0.000% |

We notice that the negligible out performance of variable sublots versus consistent sublots diminishes when setup time is lowered on the second machine. In this example, no difference is observed between using variable and consistent sublots when no setup is needed for processing sublots on machines. Therefore, in real world problems where setups are negligible, consistent sublots might be more desirable due to easy implementation in the production line.

### 4.2 Lot streaming multiple products in two-stage hybrid flow shop

The problem studied in Zhang *et al.* (2005) is also a two stage hybrid flow shop with $m$ identical machines in the first stage and a single machine in the second stage. Sublots are consistent and the number of sublots per job/product is a decision variable in that study. The authors use the rotation method to sequence sublots in the second stage. In their example, jobs 1, 2, and 3 with 8000, 10000, and 13000 identical units, respectively, are processed in two stages. The values of parameters are given in Table 4-3.

**Table 4-3 Parameter values in the example**

| | Job 1 | | Job 2 | | Job 3 | |
|---|---|---|---|---|---|---|
| | $U_1 = 8000$ | | $U_2 = 10000$ | | $U_3 = 13000$ | |
| | Processing | Setup | Processing | Setup | Processing | Setup |
| Stage 1 | 1 | 50 | 1 | 20 | 1 | 10 |
| Stage 2 | 1 | 50 | 0.5 | 5 | 0.3 | 1 |

A lower bound of 800 is assumed for sublot size to decrease computational time in large size problems. The objective function value of 12637.67 is achieved as the mean completion time of all the three jobs as illustrated in Figure 4-5.

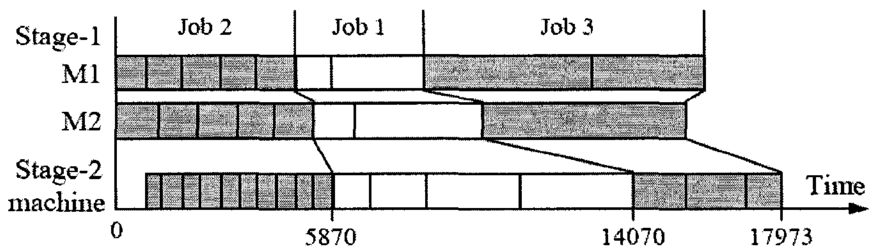



**Figure 4-5 The optimal schedule of the three jobs []**

The optimal solution schedules jobs 2, 1, and 3 in that order and processes them in 10, 4, and 3 sublots, respectively.

In this thesis we assume 10 as the maximum number of sublots per product and run the model in Section 3.2 with parameter values in this example. The same global optimum results are attained and only 4 and 3 sublots are produced for jobs 1 and 3 respectively instead of 10.

The mathematical programming model is solved to optimality in 8 minutes for this example problem. Computational time is high for larger size problems; therefore, we use the relaxed model for our further experiments as discussed in chapter 3 Section 3.3.3. As such, we determine the number of sublots per products manually.

By removing the lower bound in this example, optimal objective function value of 12617.71 is achieved. This change has increased the computational time while small size problems are still solvable.

As mentioned before, rotation method is not used in solving our model yet global optimal solutions are achieved.

### 4.2.1 Decomposing the multiple-product problem into single-product problems

Some researchers decompose the multiple-product lot streaming problem into single-product problems and solve the single product model for each product separately. In this Section we study the efficiency of such approach. It is of great importance to determine the advantage of the proposed model in this research over the single product model. As such, we run the model for each of the three products in the example one at a time. Given completion time of the last sublot of a product on each machine we can sequence the next product and continue this procedure until the last product completes production in the last stage. In the example under study we attain the optimal solution for the first product, give the required completion times to the model and solve it for the second product and so on. In other words we sequence products 1, 2, and 3 in that order manually and calculate the total makespan. Figure 4-6 shows the solution using the decomposing approach.

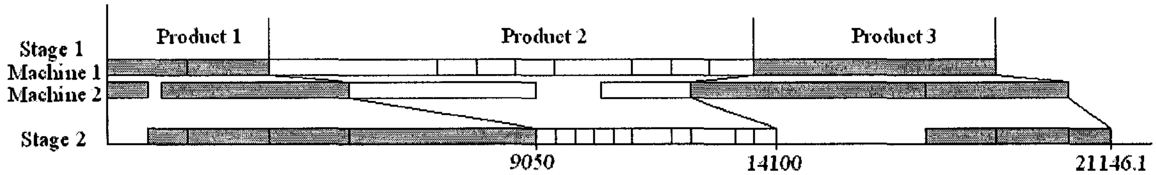



**Figure 4-6 Optimal solution for the decomposed problem**

The makespan is 21146.1 and is greater than the optimal solution, 17973 considering the three products together in the multiple-product model. It is concluded that the quality of the optimal solution is strongly related to sequencing products and the proposed model enhances the makespan in multiple-product lot streaming problems. The makespan might be even more reduced where intermingling is concerned and sublots of a product mix with those of other products.

### 4.2.2 Effects of adding parallel machines to the production line

In this step we study the effects of adding parallel machines in the two stages on the mean completion time. We perform experiments with increased number of machines in the first stage while keeping the single machine in the second stage. Consider identical parallel machines which enable overlapping of processing the parts in the first stage. The same lower bound of 800 is considered for sublot sizes and the number of sublots for products is the same 4, 10, and 3, respectively, for products 1, 2, and 3.

The rest of the assumptions hold in all the experiments. We input the values of parameters in the example and run our model in Section 3.2 for 3, 4, 5, and 6 parallel machines in the first stage. Global optimum results show that by using three and four machines in the first stage the objective function value reduces by 9.68% and 11.33% respectively. However, no further advantage is observed in using more than four machines in the first stage.

It is obvious that the bottleneck of the production is in stage 2 since rising production flow in the first stage has no impact on the objective function of the problem. Therefore, secondly we utilize parallel machines in the second stage. In this step, experiments are performed for 2, 3, and 4 parallel machines in stage two. The objective function is further improved with increased number of machines in stage two. However, as illustrated in Figure 4-7 the advantage becomes less significant when more than three machines are used in stage two. In other words, utilizing more machines in the production line may not necessarily decrease the completion time of production.

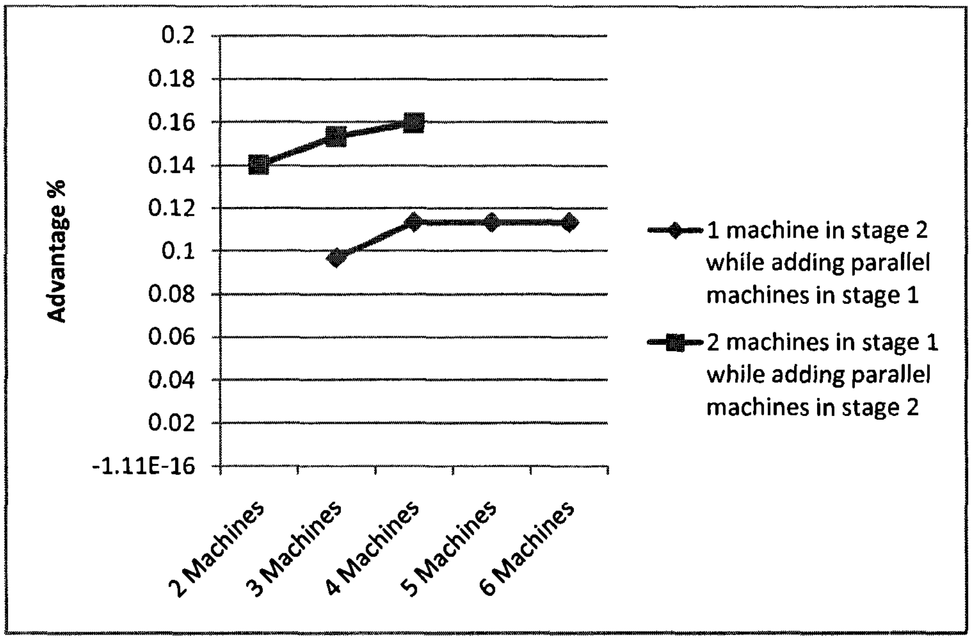



**Figure 4-7 Improvement in the objective function by adding up parallel machines**

Figure 4-8 shows the result of using two parallel machines in both stages and keeping the same number of sublots per product in the example under study. The mean completion time is reduced to 10865.77, an advantage of 14.1 %, and global optimal results are achieved. This solution is obtained in a few minutes of computation in solving the model.

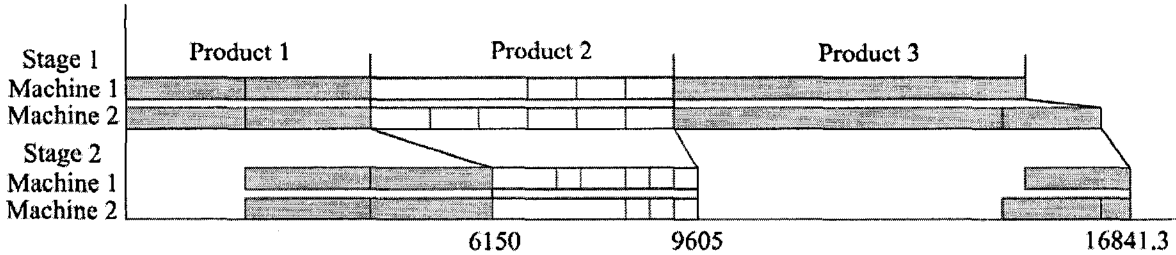



**Figure 4-8 Utilizing parallel machines in the second stage**

Figure 4-8 shows noticeable gaps in the second stage and machines are idle during processing sublots of the third product in the first stage. This situation is quite challenging in solving lot streaming problems. We intend to increase the number of sublots and carry out an experiment considering 5 consistent sublots for the third product instead of 3 sublots. The mean completion time achieved in this step is 10601.1, a global optimal solution, and the objective function is decreased by 16.12% as shown in Figure 4-9, by splitting lot and overlapping the processes, idling is less and completion time of all the products is reduced to 16048.3.

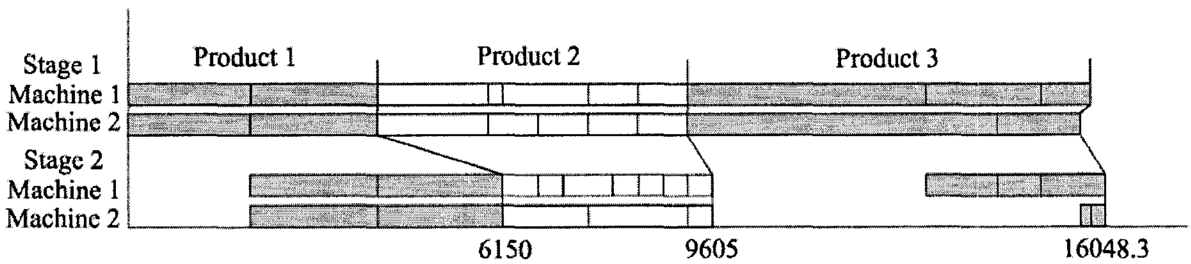



**Figure 4-9 Improved schedule by applying more sublots for the third product**

### 4.2.3 Increasing the number of sublots for each product at a time

We assume also constant number of machines in parallel stages and changed the number of sublots per product to study the effects on the objective function.

We consider two machines in the first stage and three machines in the second stage. We start from the first product and perform experiments with 3, 4, and 5 consistent sublots while the other two products have 2 sublots. We then considered the situation that products 2 and 3 have 3, 4, and 5 sublots each time with other products having 2 sublots. Experiments are performed using these settings and optimal solutions are obtained as shown in Table 4-4.

**Table 4-4 The advantages by increasing the number of sublots for each product at a time**

| Number of sublots | | | Objective | Advantage % |
|---|---|---|---|---|
| Product 1 | Product 2 | Product 3 | | |
| 2 | 2 | 2 | 12400.67 | 1.88% |
| 3 | 2 | 2 | 11892 | 5.90% |
| 4 | 2 | 2 | 11636.8 | 7.92% |
| 5 | 2 | 2 | 11528.04 | 8.78% |
| 2 | 3 | 2 | 12050 | 4.65% |
| 2 | 4 | 2 | 11839.9 | 6.31% |
| 2 | 5 | 2 | 11761.82 | 6.93% |
| 2 | 2 | 3 | 12111.8 | 4.16% |
| 2 | 2 | 4 | 11886.7 | 5.94% |
| 2 | 2 | 5 | 11857 | 6.18% |

Increasing the number of sublots normally decreases the mean completion time of production. Figure 4-10 depicts the advantage trend in the objective function value with the data given in Table 4-4.

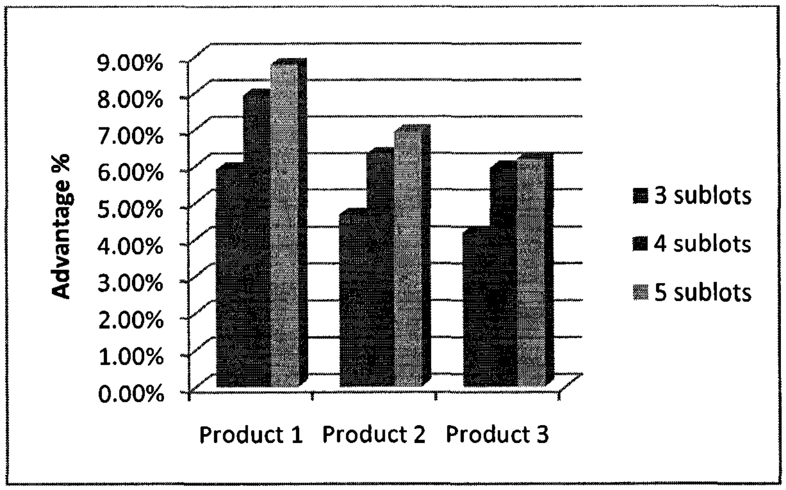



**Figure 4-10 The advantage trend in the objective value by increasing number of sublots**

## 4.3 Lot streaming multiple products in a multi-stage hybrid flow shop

In this section we use another example problem to verify the proposed integer programming model in Section 3.2. In this example, 3 products are produced in a four stage hybrid flow shop with non-identical machines. Products are transferred from stage 1 to stage 4 after completion of their process on each machine. Stage 1 and stage 4 each has a single machine while stages 2 and 3 have two and three parallel machines, respectively. In this problem, sublots are transferred along a general type of hybrid flow shop. A single machine stage can either succeed or follow a parallel machine stage. These parallel machine stages can be placed in sequence as well. The objective is to determine continuous and consistent sublot sizes so that the completion time is minimized. Table 4-5 shows the values of the parameters of this example.

**Table 4-5 Values of parameters in a four stage hybrid flow shop**

| | | Product 1 | | | | Product 2 | | | | Product 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $U_1 = 2000, SP_1 = 2$ | | | | $U_2 = 2500, SP_2 = 4$ | | | | $U_3 = 1800, SP_3 = 3$ | | | |
| | | Processing | | Setup | | Processing | | Setup | | Processing | | Setup | |
| Stage 1 | Machine1 | $P_{111}$ | 1 | $ST_{111}$ | 2 | $P_{211}$ | 2 | $ST_{211}$ | 1 | $P_{311}$ | 2 | $ST_{311}$ | 2 |
| Stage 2 | Machine1 | $P_{121}$ | 2 | $ST_{121}$ | 3 | $P_{221}$ | 2 | $ST_{221}$ | 1 | $P_{321}$ | 1 | $ST_{321}$ | 2 |
| | Machine2 | $P_{122}$ | 2 | $ST_{122}$ | 3 | $P_{222}$ | 2 | $ST_{222}$ | 1 | $P_{322}$ | 1 | $ST_{322}$ | 2 |
| Stage 3 | Machine1 | $P_{131}$ | 3 | $ST_{131}$ | 4 | $P_{231}$ | 4 | $ST_{231}$ | 2 | $P_{331}$ | 2 | $ST_{331}$ | 3 |
| | Machine2 | $P_{132}$ | 2 | $ST_{132}$ | 3 | $P_{232}$ | 3 | $ST_{232}$ | 1 | $P_{332}$ | 1 | $ST_{332}$ | 2 |
| | Machine3 | $P_{133}$ | 3 | $ST_{133}$ | 4 | $P_{233}$ | 4 | $ST_{233}$ | 2 | $P_{333}$ | 2 | $ST_{333}$ | 3 |
| Stage 4 | Machine1 | $P_{141}$ | 0.5 | $ST_{141}$ | 1 | $P_{241}$ | 1 | $ST_{241}$ | 0.5 | $P_{341}$ | 1 | $ST_{341}$ | 1 |

Machines in stage 2 are identical while the third stage has different machines with different processing and setup times. Assume that a factory has bought a new processing machine which works faster with less setup time. Since the goal is to minimize the completion time of all products, it is logical to use the faster machine as much as possible. In our example, it is expected that more sublots are assigned to machine 2 in the third stage.

We input the parameter values and solve the integer programming model by Lingo. Global optimal results are obtained in few minutes and sublot sizes are shown in Table 4-6.

**Table 4-6 Size of consistent continuous sublots**

| Product 1 | | Product 2 | | | | Product 3 | | |
|---|---|---|---|---|---|---|---|---|
| Size of sublots | | Size of sublots | | | | Size of sublots | | |
| 1 | 2 | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| 992.12 | 1007.88 | 823.45 | 617.4 | 540.17 | 518.98 | 666.30 | 647.97 | 485.73 |

As seen in the Gant chart shown in Figure 4-11, sublots of different products do not mix; they start processing as the predecessor product has completed its process. Second machine of the third stage has been assigned more sublots for processing as expected. Machines 1 and 4r have less idle times and production flow is slow on them.

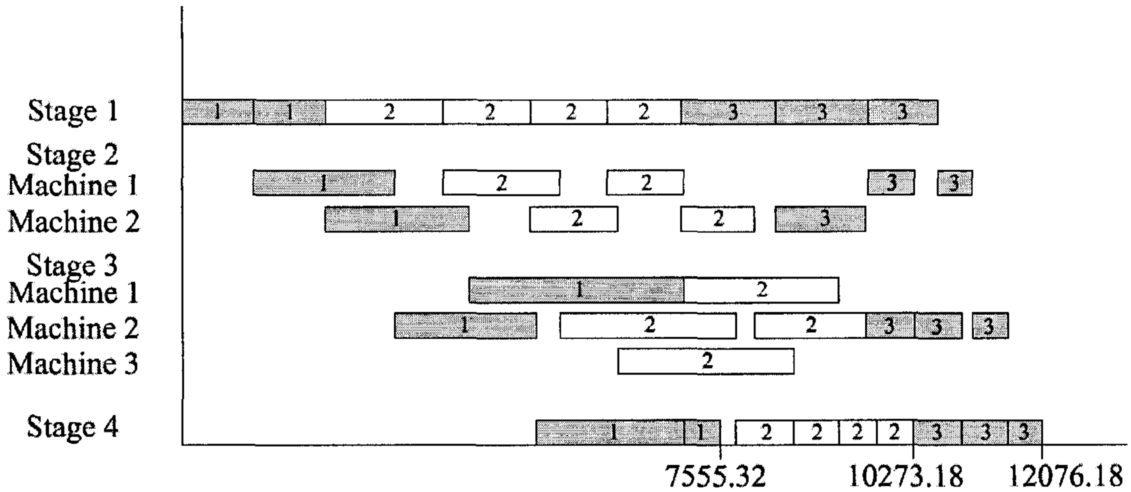



**Figure 4-11 Final schedule of the hybrid flow shop lot streaming example**

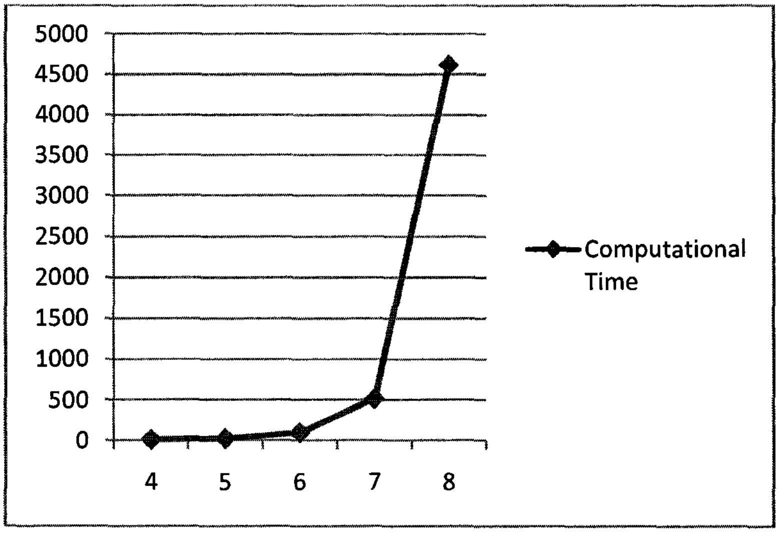
The minimum completion time achieved is 12076.18 and the optimal sequence is product 1, product 2, and product 3 consecutively.

It is of great interest to study which combination of parallel machines in the stages can yield the best response in solving hybrid flow shop lot streaming problems. We consider all the possible combinations of machines in the four stages in our example with maximum number of 2 machines in each stage. Where needed, another identical machine is utilized in stages 1 and 4. In the third stage, machine 3 is removed and machine 1 is kept when only one machine is used. The rest of the assumptions hold and parameter values are input in our model to perform the experiments. Sixteen possible combinations are illustrated in Table 4-7. The model is run in Lingo and global optimal solutions are achieved. Table 4-7 shows the corresponding makespan, computational time and sequence of products for each alternative.

**Table 4-7 Different combinations of machines and the responses**

| No. | Number of machines in stages | | | | Makespan | Time (Seconds) | Sequence of Products |
|---|---|---|---|---|---|---|---|
| | Stage 1 | Stage 2 | Stage 3 | Stage 4 | | | |
| 1 | 1 | 1 | 1 | 1 | 20994.18 | 12 | 2-3-1 |
| 2 | 1 | 1 | 1 | 2 | 20976.25 | 14 | 2-3-1 |
| 3 | 1 | 1 | 2 | 1 | 13704.83 | 27 | 1-2-3 |
| 4 | 1 | 2 | 1 | 1 | 20994.18 | 26 | 2-3-1 |
| 5 | 2 | 1 | 1 | 1 | 20532.27 | 31 | 2-3-1 |
| 6 | 1 | 1 | 2 | 2 | 12860.66 | 31 | 1-2-3 |
| 7 | 1 | 2 | 1 | 2 | 20976.25 | 46 | 2-3-1 |
| 8 | 1 | 2 | 2 | 1 | 20994.18 | 38 | 2-3-1 |
| 9 | 2 | 1 | 1 | 2 | 20514.35 | 77 | 2-3-1 |
| 10 | 2 | 1 | 2 | 1 | 13538.75 | 170 | 1-3-2 |
| 11 | 2 | 2 | 1 | 1 | 20532.27 | 214 | 2-3-1 |
| 12 | 1 | 2 | 2 | 2 | 11851.39 | 408 | 1-2-3 |
| 13 | 2 | 1 | 2 | 2 | 12773.17 | 499 | 2-1-3 |
| 14 | 2 | 2 | 1 | 2 | 20514.35 | 550 | 2-3-1 |
| 15 | 2 | 2 | 2 | 1 | 10610.59 | 600 | 2-3-1 |
| 16 | 2 | 2 | 2 | 2 | 10335.86 | 4620 | 2-3-1 |

This model is solved within few seconds for small size problems and up to 77 minutes for problems with 2 parallel machines in each of the four stages; this setting yields the minimum makespan out of all. As illustrated in Figure 4-12, computational time of solving the model increases by the cumulative number of machines in stages. In this figure, the computational time is taken as the average of the time corresponding to settings with the same number of machines.




**4-12 Increase of computational time by the number of machines**

By analyzing the results, we can conclude that using more machines in particular stages may not necessarily decrease the completion time of products. Setting 1, which is a pure flow shop, and setting 8 have the same objective value therefore the combination of 1, 2, 2, and 1 machine in stages 1, 2, 3, and 4 respectively can be disregarded. Moreover, setting 16 has the minimum makespan with the highest computational time. Taking into account all these alternatives in hybrid flow shop problems enables the management to balance between costs and time in manufacturing systems.

### 4.4 Summary

The mixed integer programming formulation developed in Chapter Three is programmed and run using Lingo optimization software. The data used in the experiments is either hypothetical or from the similar example problems in the literature. The model is used to solve lot streaming problems in multi-stage hybrid flow shops. The obtained results are reasonable and the performed comparisons to the similar existing studies validate the proposed model. The solution approach is adequate for solving problems with various settings.

In this chapter, insights are gained into the effects of changing different model parameters and problem characteristics on the objective function. Experiments are performed and problems are solved giving parameter values in three different example problems. Changing the number of machines in each stage, the number of sublots per product, and type of sublots are discussed and influences on the makespan are analyzed.

From the first example problem in Section 4.1, we observe that by applying variable sublots instead of consistent and constant sublots, the makespan is reduced due to high setup time in stage 2. It also shows that when setup time is negligible, there is no difference between using consistent sublots and variable sublots. This is studied in another example problem considering a single product in a hybrid flow shop of three parallel machines in the first stage and a single machine in the second stage.

Some researchers decompose the multiple-product lot streaming problem into single-product problems and solve them separately. In the example problem in Section 4.2, we observe that this approach increases the makespan. The best solution may be achieved by sequencing decision and sublot size determination simultaneously in a multiple-product model. Moreover, influences of increasing the number of sublots each at a time on the makespan are analyzed.

Further analysis is carried out in an example problem of lot streaming multiple products in a multi-stage hybrid flow shop in Section 4.3. The results show that the optimal solution assigns more sublots to the faster machine among the parallel machines in the third stage, as expected. This model incorporates different combinations of non-identical parallel machines or single machine in different stages.

# Chapter Five Conclusions and Future Research

This chapter presents a summary of the research conducted in this thesis. Several concluding remarks are also presented based on the problem modeling and results analysis. Future research directions on this study are also discussed.

## 5.1 Concluding Summary

This research extends the work of Biskup and Feldmann (2006) and Feldmann and Biskup (2006). They present integer programming formulations for lot streaming problem of single product and multiple products in flow shops. Zhang et al. (2005) and Liu (2008) develop mathematical and heuristic methods to solve the problem of lot streaming in two-stage hybrid flow shops for single and multiple products. This research incorporates lot streaming multiple products in multi-stage hybrid flow shops. A mixed integer programming model is proposed solve this problem. We assume consistent sublots that do not intermingle while processing on machines. Machines are non-identical and setups on the machines are independent of the sequence of sublots. The solution approach determines the number and the size of sublots as well as product sequence so that the makespan is minimized. The presented model is capable of handling lot streaming problems with various settings; nevertheless, it may not yield optimal results for large size problems in short computational time.

Numerical experiments are carried out to validate the proposed model. Problem characteristics and parameter values are varied to analyze their influences on the objective function value. Comparisons are performed to similar example problems in the studied literature. It is illustrated that using parallel machines in some stages of the production line may not necessarily reduce the objective function value. Generated optimal solutions verify the accuracy of the proposed mixed integer linear programming model. The presented approach can help the shop manager to decide the best combination of parallel and single machines corresponding to the best objective function value in hybrid flow shops.

**5.2 Future directions for research**

Although this research provides interesting and useful results, the underlying research possibilities in lot streaming problems exist. Some of the possible extensions to this work include:

- Developing a mathematical model for solving the multiple-product lot streaming problem in multi-stage hybrid flow shops considering variable sublots
- Considering sequence-independent setup time
- Considering intermingling
- Evaluating the efficiency of decomposition approach when intermingling is allowed
- Developing heuristic methods to decrease computational time of solving large size lot streaming problems

# References


1. Allahverdi A., Gupta J.N.D., Aldowaisan T., 1999. A review of scheduling research involving setup considerations, Omega, **27**, 219-239.
2. Baker K.R., 1995. Lot streaming in the two-machine flow shop with setup times, Annals of Operations Research, **57**, 1-11.
3. Biskup D., Feldmann M., 2006. Lot streaming with variable sublots: an integer programming formulation, Journal of the Operational Research Society, **57**, 296–303.
4. Bukchin J., Tzur M., Jaffe M., 2002. Lot splitting to minimize average flow-time in a two-machine flow-shop, IIE Transactions, **34**, 953-970.
5. Cetinkaya F.C., and Kayaligil M.S., 1992. Unit sized transfer batch scheduling with setup times, Computers and Industrial Engineering, **22**, 177-183.
6. Cetinkaya F.C., 1994. Lot streaming in a two-stage flow shop with set-up, processing and removal times separated, Journal of Operational Research Society, **45**, 1445-1455.
7. Chang J.H., and Chiu H.N., 2005. A comprehensive review of lot streaming, International Journal of Production Research, **43**, 1515–1536.
8. Chen J. and Steiner G., 1997. Approximation methods for discrete lot streaming in flow shops, Operations Research Letters, **21**, 139-145.
9. Chen J. and Steiner G., 2003. On discrete lot streaming in no-wait flow shops, IIE Transactions, **35**, 91-101.
10. Chen J., and Steiner G., 1998. Lot streaming with attached setups in three-machine flow shops, IIE Transactions, **30**, 1075-1084.

11. Chen J. and Steiner G., 1999. Discrete lot streaming in two-machine flow shops, INFOR, **37**, 160-173.

12. Chiu H.N., Chang J.H., and Lee C.H., 2004. Lot streaming models with a limited number of capacitated transporters in multistage batch production systems, Computers & Operations Research, **31**, 2003-2020.

13. Chiu H.N., and Chang J.H., 2005. Cost models for lot streaming in a multistage flow shop, Omega, **33**, 435-450.

14. Feldmann M., and Biskup D., 2006. Lot streaming in a multiple product permutation flow shop with intermingling, International Journal of Production Research, 1-20.

15. Gupta J.N.D., and Tunc E.A., 1991. Schedules for a two-stage hybrid flowshop with parallel machines at the second stage, International Journal of Production Research, **29**, 1489-1502.

16. Kalir A.A., and Sarin S.C., 2001. A near-optimal heuristic for the sequencing problem in multiple-batch flow-shops with small equal sublots, Omega, **29**, 577-584.

17. Kalir A.A. and Sarin S.C., 2001. Optimal solutions for the single batch, flow shop, lot-streaming problem with equal sublots, Decision Sciences, **32**, 387-397.

18. Kher H.V., Malhotra M.K., and Steele D.C., 2000. The effect of push and pull lot splitting approaches on lot traceability and material handling costs in stochastic flow shop environments, International Journal of Production Research, **38**, 141-160.

19. Kropp D.H., and Smunt T.L., 1990. Optimal and heuristic solutions for lot splitting in a flow shop, Decision Sciences, **21**, 691-709.

20. Lee C.Y., Lei L. and Pinedo M., 1997. Current trends in deterministic scheduling, Annals of Operations Research, **70,** 1–41.

21. Liu J., 2008. Single-job lot streaming in $m$ - 1 two-stage hybrid flowshops, European Journal of Operational Research, **187**, 1171-1183.

22. Logendran R., and Subur F., 2004. Unrelated parallel machine scheduling with job splitting, IIE Transactions, **36**, 359-372.

23. Lundrigan R., 1986. What Is This Thing Called OPT?, Production and Inventory Management Journal, **27**, 2-11.

24. Potts C.N. and Van Wassenhove L.N., 1992. Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity, Journal of Operational Research Society, **43**, 395-406.

25. Reiter S., 1966. A system for managing job-shop production, Journal of Business, **39**, 371–393.

26. Riane F., Artiba A., and Elmaghraby S.E., 2002. Sequencing a hybrid two-stage flowshop with dedicated machines, International Journal of Production Research, **40**, 4353-4380.

27. Ruiz R., and Maroto C., 2006. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility, European Journal of Operational Research, **169**, 781-800.

28. Sen A. and Benli O.S., 1999. Lot streaming in open shops, Operations Research Letters, **23**, 135-142.

29. Sriskandarajah C. and Wagneur E., 1999. Lot streaming and scheduling multiple products in two-machine no-wait flowshops, IIE Transactions, **31**, 695–707.

30. Sriskandarajah C., and Wagneur E., 1999. Lot streaming and scheduling multiple products in two- machine no-wait flowshops, IIE Transactions, **31**, 695-707.

31. Subodha K., Bagchi T.P., and Sriskandarajah C., 2000. Lot streaming and scheduling heuristics for m-machine no-wait flowshops, Computers and Industrial Engineering, **38**, 149-172.

32. Trietsch D., Baker K.R., 1993. Basic techniques for lot streaming, Operations Research, **41**, 1065-1076.

33. Vickson R.G., and Alfredsson B.E., 1992. Two- and three-machine flow shop scheduling problems with equal sized transfer batches, International Journal of Production Research, **30**, 1551-1574.

34. Vickson R.G., 1995. Optimal lot streaming for multiple products in a two-machine flow shop, European Journal of Operational Research, **85**, 556-575.

35. Yalaoui F., and Chu C., 2003. An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times, IIE Transactions, **35**, 183–190.

36. Yoon S.H, and Ventura J.A., 2002. Minimizing the mean weighted absolute deviation from due dates in lot-streaming flow shop scheduling, Computers & Operations Research, **29**, 1301-1315.

37. Yoon S.H, and Ventura J.A., 2002. An application of genetic algorithms to lot-streaming flow shop scheduling, IIE Transactions, **34**, 779-787.

38. Zhang W., Yin C., Liu J., and Linn R.J., 2005. Multi-job lot streaming to minimize the mean completion time in *m*-1 hybrid flowshops, International Journal of Production Economics, **96**, 189-200.

# Appendix A: Lingo Code for Multiple-Product Model

```
SETS:
product/1..3/:U, SP;                                  !j,k;
sublot/1..4/;                                         !i,t,r;
stage/1..4/:SM;                                             !s;
machine/1..3/;                                        !m,f;
prodmach(product,stage,machine):P,ST;                 !j,s,m;
compsize(product,sublot,stage,machine):C,L,B;         !j,i,s,m;
prosub(product,sublot,product,sublot):X;              !j,i,k,t;
link(sublot,machine);
Link2(product,sublot,stage);

ENDSETS
DATA:
U = 2000 2500 1800;
SM = 1 2 3 1;
SP = 2 4 3;
G = 10000000;

END DATA

!OBJECTIVE FUNCTION;

MIN= Z;

!SUBJECT TO;

!#1;
@FOR(compsize(j,i,s,m)|i #LE# SP(j) : Z >= C(j,i,4,m));

!#2;
@FOR(product(j):@for(stage(s): @SUM(link(i,m)|i #LE# SP(j):L(j,i,s,m))=U(j)));

!#3;
@FOR (compsize(j,i,s,m)|m #GT# SM(s) #AND# i #LE# SP(j):
                              L(j,i,s,m)=0 );
!#4;
@FOR(compsize(j,i,s,m) |i #LE# SP(j): C(j,i,s,m)>= P(j,s,m)*L(j,i,s,m)+
ST(j,s,m)*B(j,i,s,m) );

!#5;
@FOR(Compsize(j,i,s,m)|i #LE# SP(j):@for(sublot(r)|r #LT# i #AND# r #LE# SP(j):
                        C(j,i,s,m)-P(j,s,m)*L(j,i,s,m)-ST(j,s,m)*B(j,i,s,m)>= C(j,r,s,m)));
```

```
!#6;
@FOR(machine(m):  @for(stage(s):
        @for(Prosub(j,i,k,t)|j#LT#k #AND# i #LE# SP(j)#AND# t #LE# SP(k):
                C(k,1,s,m)-P(k,s,m)*L(k,1,s,m)-ST(k,s,m)*B(k,1,s,m) >= C(j,i,s,m) - (1-
X(j,i,k,t))*G)));

!#7;
@FOR(machine(m):@for(stage(s):
        @for(Prosub(j,i,k,t)|j#LT#k #AND# i #LE# SP(j)#AND# t #LE# SP(k):
                C(j,1,s,m)-P(j,s,m)*L(j,1,s,m)-ST(j,s,m)*B(j,1,s,m) >= C(k,t,s,m) -
X(j,i,k,t)*G)));

!#8;
@FOR(Compsize(j,i,s,m)|s#GT#1 #AND# i #LE# SP(j):
                @for(machine(f): C(j,i,s,m)-P(j,s,m)*L(j,i,s,m)-ST(j,s,m)*B(j,i,s,m)>=
C(j,i,s-1,f)));

!#9;
 @FOR(Prosub(j,i,k,t)|i#GT#1 #AND# t#GT#1 #AND# i #LE# SP(j)#AND# t #LE#
SP(k) #AND# j#LT#K :
                                X(j,i,k,t) <= X(j,i-1,k,t-1));

!Control of the binary B;
!#10;
@FOR (link2(j,i,s)|i #LE# SP(j):
                        @sum(machine(m):B(j,i,s,m))=1);
!#11;
@FOR (compsize(j,i,s,m)|i #LE# SP(j): L(j,i,s,m) <= B(j,i,s,m)*G );

!Consistent;
!#12;
@FOR (link2(j,i,s)|s #GT# 1 #AND# i #LE# SP(j):
                        @sum(machine(m):L(j,i,s,m))=@sum(machine(m):L(j,i,s-1,m)));

!#13;
@FOR(Prosub(j,i,k,t)|j#LT#k: @BIN(X(j,i,k,t)));

!#14;
@FOR (compsize(j,i,s,m): @BIN(B(j,i,s,m)));
```

# Appendix B: Lingo Code for the Model with Variable Sublots

```
SETS:
sublot/1..2/;                                     !i,t,u;
stage/1..2/:stam;                                           !s;
machine/1..2/;                                    !m,f;
process(stage,machine):p,st;                  !s,m;
compsize(sublot,stage,machine):C,L,B;         !i,s,m;
submasub(sublot,stage,sublot):y;              !i,s,t;
link(sublot,machine);
Link2(sublot,stage);

ENDSETS
DATA:
U=4000;
P =    1 1
       0 0;
st = 0 0 2000 2000;
stam= 2 1;
G=1000000;

END DATA

!OBJECTIVE FUNCTION;

MIN=Z;

!SUBJECT TO;

!#1;
@FOR(compsize(i,s,m): Z >= C(i,2,m));

!#2;
@FOR(stage(s): @SUM(link(i,m):L(i,s,m))=U);

!#3;
@FOR (compsize(i,s,m)|m #GT# stam(s):L(i,s,m)=0 );

!#4;
@FOR(compsize(i,s,m) : C(i,s,m)>= P(s,m)*L(i,s,m)+ st(s,m)*B(i,s,m) );

!@FOR(compsize(i,s,m) : C(1,s,m)>= P(s,m)*L(1,s,m)+ st(s,m)*B(1,s,m) );

!#5;
@FOR(Compsize(i,s,m):@for(sublot(r)|r #LT# i:
                  C(i,s,m)-P(s,m)*L(i,s,m)-st(s,m)*B(i,s,m)>= C(r,s,m)));
```

```
!#6;
@FOR(Compsize(i,s,m)|s#GT#1:@for(sublot(t):
            @for(machine(f): C(i,s,m)-P(s,m)*L(i,s,m)-st(s,m)*B(i,s,m)>= C(t,s-1,f)-
(1-y(i,s,t))*G)));
!Control of the binary B;
!#7;
@FOR (link2(i,s):
                @sum(machine(m):B(i,s,m))<=1);
!#8;
@FOR (compsize(i,s,m): L(i,s,m) <= B(i,s,m)*G );

!#9;
@FOR (submasub(i,s,t)| s #GT# 1 :
        y(i,s,1)=1 );
!#10;
@FOR (submasub(i,s,t)|t#LE#1 #and# s#GT#1:
                y(i,s,t+1)<= y(i,s,t));

!#11;
@FOR (compsize(i,s,m)| s #GT# 1 :
@FOR (sublot(t)|t#LE#1:
        L(i,s,m) <= @sum(link(u,m)|u #LE# t: L(u,s-1,m))- @sum(link(u,m)| u#LE#i-
1: L(u,s,m))
            + (1 - y(i,s,t) + y(i,s,t+1))* G ));

!Consistent;
!#12;
@FOR (link2(i,s)|s #GT# 1: @sum(machine(m):L(i,s,m))=@sum(machine(m):L(i,s-
1,m)));
!Constant;
@FOR (link2(i,s)|i #GT# 1: @sum(machine(m):L(i,s,m))=@sum(machine(m):L(i-
1,s,m)));

!#13;
@FOR (compsize(i,s,m):@BIN(B(i,s,m)));

!#14;
@FOR (submasub(i,s,t):@BIN(y(i,s,t)));
```