MATHEMATICAL MODELS AND HEURISTIC ALGORITHMS FOR A MULTI-PRODUCT LOT STREAMING PROBLEM IN A TWO-MACHINE FLOWSHOP

ŞAHİKA AKDOĞAN

FEBRUARY 2017

MATHEMATICAL MODELS AND HEURISTIC ALGORITHMS

FOR

A MULTI-PRODUCT LOT STREAMING PROBLEM

IN

A TWO-MACHINE FLOWSHOP

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF

NATURAL AND APPLIED SCIENCES

OF

ÇANKAYA UNIVERSITY

BY

ŞAHİKA AKDOĞAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF MASTER OF SCIENCE

IN

THE DEPARTMENT OF

MATHEMATICS AND COMPUTER SCIENCE

Title of the Thesis: **Mathematical Models and Heuristic Algorithms for a Multi-Product Lot Streaming Problem in a Two Machine Flowshop**

Submitted by **Şahika AKDOĞAN**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Halil T. EYYUBOĞLU
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Fahd JARAD
Head of Department

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

Assoc. Prof. Dr. Ferda Can ÇETİNKAYA
Supervisor

Examination Date: 06.02.2017

Examining Committee Members

Assoc. Prof. Dr. Ferda Can ÇETİNKAYA (Çankaya Univ.)

Asst. Prof. Dr. Özlem DEFTERLİ (Çankaya Univ.)

Asst. Prof. Dr. Bülent Gürsel EMİROĞLU (Kırıkkale Üniv.)

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name	**Şahika**
Last Name	**Akdoğan**
Signature	
Date	06.02.2017

ABSTRACT

MATHEMATICAL MODELS AND HEURISTIC ALGORITHMS FOR A MULTI-PRODUCT LOT STREAMING PROBLEM IN A TWO MACHINE FLOWSHOP

AKDOĞAN, Şahika
M.Sc., Department of Mathematics and Computer Science
Supervisor: Assoc. Prof. Dr. Ferda Can ÇETİNKAYA
February 2017, 100 pages

In this study, we consider a multi-product lot streaming problem to minimize the makespan on a two-machine flowshop environment in which all product lots are processed by Machine 1 and then by Machine 2. Most of the current studies in the literature of the multi-product lot streaming problem assume that the number of sublots for each product is known in advance, and determines the size for each sublot of every product and the sequence of sublots of all products. However, in our study we assume that the total number of sublots for all products is known advance, although the number of sublots for each product is not known in advance. Our problem is to determine the number of sublots for each product, the size of each sublot and the sequence of sublots that gives the minimum makespan. We investigate this multi-product lot streaming problem for two cases in which sublots of each product are equal sized in the first case while sublots of each product are unequal sized in the second case. We develop mixed integer linear mathematical models and heuristic algorithms for solving each case. We compare these solutions of mathematical models and heuristic algorithm. We design computational experiments to evaluate the performance of our solutions approaches in terms of makespan time. The results show that the mixed integer programming models do not seem to be a useful alternative, especially for large scale problem instances. However, our proposed heuristic algorithms find optimal or near-optimal solutions in very short time.

Keywords: Lot Streaming, Equal Sublots, Unequal Sublots, Makespan

ÖZ

İKİ MAKİNALI AKIŞ TİPİ ATÖLYEDE
ÇOK ÜRÜNLÜ KAFİLE BÖLME VE KAYDIRMA PROBLEMİ İÇİN
MATEMATİKSEL MODELLER
VE
SEZGİSEL ALGORİTMALAR

AKDOĞAN, Şahika
Yüksek Lisans, Matematik-Bilgisayar Bölümü
Tez Yöneticisi: Doç. Dr. Ferda Can ÇETİNKAYA
Şubat 2017, 100 sayfa

Bu çalışmada, tüm ürünlerin önce birinci ve daha sonra ikinci makinede işlem gördüğü iki makinalı bir akış tipi üretim sisteminde tüm ürün kafilelerinin bitirilme süresini en küçükleyen çok ürünlü kafile bölme ve kaydırma problemi ele alınmıştır. Çok ürünlü kafile bölme ve kaydırma problemi literatüründe yer alan çalışmaların çoğu, her ürün kafilesinde yer alan alt kafilelerin sayısının önceden bilindiğini varsayar ve her ürüne ait alt kafilelerin büyüklüğü ile tüm kafilelere ait alt kafilelerin kendi aralarındaki işlem sırasını belirler. Oysa ki, yaptığımız çalışmada, her ürün kafilesinde yer alan alt kafilelerin sayısının önceden bilinmemesine karşın tüm ürün kafilelerinde yer alan alt kafilelerin toplam sayısının önceden bilindiğini varsaymaktayız. Sorunumuz, her ürün kafilesinde yer alacak alt kafilelerin sayısını, bu alt kafilelerin büyüklüğünü ve tüm ürün kafilerinin bitirilme süresini en küçükleyecek şekilde tüm kafilelere ait alt kafilelerin kendi aralarındaki işlem sırasını belirlemektir. Bu çok ürünlü kafile bölme ve kaydırma problemini iki farklı durum için irdeledik. Birinci durumda her ürün kafilesindeki alt kafileler eşit büyüklükteyken, ikinci durumda her ürün kafilesindeki alt kafileler eşit olmayan büyüklükte olabilmektedir. Çözüm yaklaşımlarımızın hem çözüm kalitesi hem de süresi açısından performansını değerlendirmek için sayısal deneyler tasarladık. Sonuçlar, karışık tam sayılı programlama modellerinin özellikle büyük ölçekli problem örnekleri için yararlı bir alternatif olmadığını göstermiştir. Bununla birlikte, önerdiğimiz sezgisel algoritmalar çok kısa sürede optimum veya optimuma yakın çözümler bulmaktadır.

Anahtar Kelimeler: Kafile Bölme ve Kaydırma, Eşit Alt Kafileler, Eşit Olmayan Alt Kafileler, Tüm Ürünlerin Bitirilme Süresi

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Assoc. Prof. Dr. Ferda Can ÇETİNKAYA for his supervision, special guidance, suggestions, encouragement, and patience through the development of this thesis.

It is a pleasure to express my special thanks to my family and my friends for their priceless support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

CONWIP	Constant Work-in-Process
JIT	Just-in-Time
MILP	Mixed-integer linear programming
MPLS	Multi-Product Lot Streaming
MFT	Mean Flow Time
MRP	Material Requirements Planning
MRPII	Manufacturing Resource Planning
OPT	Optimized Production Technology
UI	User Interface
WIP	Work-in-Process

CHAPTER 1

INTRODUCTION

Nowadays, to stay competitive in the industrial world market, manufacturing companies have to run an efficient operation for changing market needs. Thus, accelerated but effective methodologies of production scheduling become a key issue. Due to the batch production nature of such an environment, the use of appropriate production lot size/sizes on the shop floor is central to achieving this objective. One technique that can effectively influence the flow of a lot of jobs over the machines by appropriately determining the size of production lots, also called sublots, is *lot streaming*.

Literately, the term *lot streaming* shall be introduced as follows [1]:

"Lot streaming denotes the techniques of splitting given jobs, each consisting of identical items, into sublots to allow overlapping of successive operations in multi-stage manufacturing systems, to reduce production makespan. More specifically,

The goal of lot streaming is to determine the number of sublots for each product, the size of each sublot and the sequence for processing the sublots to minimize production makespan with all required constraints satisfied."

To make definition clear, we consider the scenario that discrete and identical products (called *lot*) are to be processed on several machines as a flow shop. Instead of transferring the entire lot, it is considered to transferring the items of the lot in smaller batches (called *sublots*). This technique of splitting lots in to sublots and processing different sublots simultaneously over different machines is called *lot* *streaming*. Briefly, lot streaming is a technique to accelerate the processing of the product when reducing the process time.

As an illustration of the lot streaming problem, suppose a lot consist of 100 items and it is processed on two machines Machine 1 (M1) and Machine 2 (M2). Suppose, the processing times per item of the lot on M1 and M2 are 2 and 1 time units, respectively. If the lot is not to be split into sublots, the distribution of the lot for processing over the machines will be as shown in Figure 1.1.

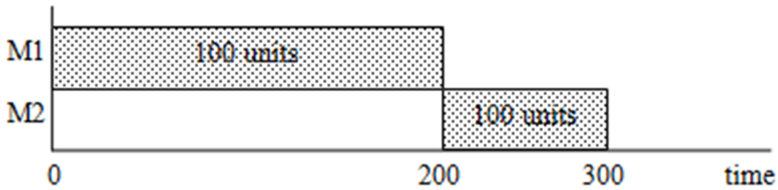


Figure 1.1 Processing without Lot Streaming

On the other hand, if the lot is split into four sublots with sizes 40, 20, 10 and 30 items and these sublots were processed in an overlapping fashion, the distribution of the lot for processing over the machines will be as shown in Figure 1.2.

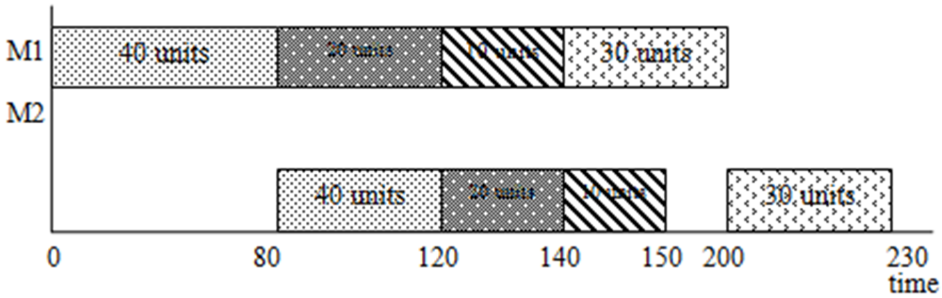


Figure 1.2 Processing with Lot Streaming

In industry, the quality and success of operations are evaluated basically three commonly used performance measures [2]. These are Makespan ($C_{\max}$), Mean Flow Time (MFT) and Average Work in Progress Levels ($\overline{WIP}$ Levels).

Makespan is defined as completion time of the last sublot on the last machine. As shown in Figure 1.1 and Figure 1.2, the reduction in makespan is obvious. However this advantage may not be that obvious if setup and/or transfer times are considered [3]:

"One apparent advantage of lot streaming is reduction in the makespan value. However, this advantage may not be that obvious if set up and/or transfer times are encountered during the handling of individual sublots. The problem get even more interesting depending on whether it can be performed a priori, i.e., before the arrival of a sublot on a machine. Also, if more than one lot is to be processed on the machines, the makespan value will depend on whether or not the sublots from different lots are intermingled. The sequence in which the lots themselves are processed can impact the makespan value as well."

Optimal production management aims to eliminate the waste created by the manufacturing system. Reducing WIP and mean flow time of the production batches are the core concepts of lean manufacturing [4]. Keeping unnecessary inventory causes a capital expense. This waste of capital is reduced to a large extend by employing the concept of lot streaming. Also waste of time is decreased by the lot streaming concept, since the main drive to apply lot streaming is to lower the makespan and MFT.

In this study, we consider a multi-product lot streaming problem on a two-machine flowshop environment in which all products (lots) are processed by Machine 1 and then by Machine 2. That is, the first and second operations of the products are performed by Machine 1 and Machine 2, respectively. Current studies in the literature assumes that the total number of sublots for each product is known in advance and the sizes of sublots of each product is to be optimally determined within the limit of the total number of sublots of the product. Our main difference from the current studies is that we assume that the total number of sublots for all products is known advance and our problem is to determine the number of sublots for each product, the size of each sublot and the sequence of sublots that gives the minimum makespan. We investigate the multi-product lot streaming problem for both equal and unequal sized sublots cases. For this purpose, we develop mixed integer linear mathematical models and heuristic algorithms for solving each case and compare these mathematical models with heuristic algorithms.

The remainder of this report is as follows. Chapter 2 provides detailed background information and review the related literature about the lot streaming problems. We briefly define our problem along with assumptions for its two cases and provide details of mathematical models in Chapter 3. The details of our proposed heuristic algorithms are explained in Chapter 4. We also provide numerical examples for our heuristic algorithms in this chapter. Chapter 5 explains the software implementation details of the proposed heuristic algorithms and provides a software usage manual. In Chapter 6, we discuss the results of our computational experiments done for determining the performance of the heuristic algorithms by comparing them with the mathematical programming models providing the optimal solution. Finally, a brief summary and conclusion of our research and future research directions are given in Chapter 7.

CHAPTER 2

LOT STREAMING: BASICS AND LITERATURE REVIEW

In this chapter, we first prove the basics of the lot streaming problem and then provide a review of recent studies on lot streaming in scheduling problems. We will discuss the related literature in two categories: single-product lot streaming and multi-product lot streaming.

2.1 Basics of Lot Streaming

In the last sixty years, thousands of papers were released about flowshop scheduling and its several variations. Especially, at the end of last century, researches were focused on a scenario where the lots are split into sublots, that is called *lot streaming.* In these studies related to the lot streaming problem, generally the goal is to determine the number of sublots for each product lot, the size of each sublot and the processing sequence of the sublots and product lots.

To understand these studies, the components of lot streaming problem must be clearly identified. The components which are derived from Chang and Chui and Feldman and Biskup are summarized in Table 2-1. [5]

Table 2-1 Components of Lot Streaming Problems

Dimension	Level	
Product Type	*Single-product	*Multi-product
Production Type	*Flow shop *Job shop	*Open shop *Arborescent shop
Sublot Type	*Fix *Equal	*Consistent *Variable
Divisibility of the Sublot Size	*Discrete	*Continuous
Sequence of the Sublots	*Intermingling	*Non-Intermingling
Operation Continuity	*Idling	*No Idling
Transfer Timing	*No-wait schedules	*Wait schedules
Performance Measures	Time models	*Makespan *Mean flow time *Total flow time *Mean tardiness *Number of tardy jobs *Total deviation from due date
	Cost models	*Total cost *Total cost with makespan
Activities Involved	Setup	*No setup *Attached setup *Detached setup
	Production	*Raw materials *Work-In-Process *Finished goods
	Transportation	*Transportation Time *Return Time *Capacities of transporters *Number of transporters

Below, we give briefly explains the terms in Table 2-1.

Product Type

- Single-product/Multiple Products: This approach considers either a single-product or a multiple products.

Production Type

- Flow Shop: In this production type, jobs are processed according to a sequence. If all the jobs follow the same route, this manufacturing system called *flow shop*.
- Job Shop: In this production type, jobs may follow different routes. They may visit the same machine once or more. This manufacturing system is called *job shop*.
- Open Shop: The *open shop scheduling problem* is a scheduling problem in which a given set of jobs must each be processed for given amounts of time at each of a given set of workstations, in an arbitrary order, and the goal is to determine the time at which each job is to be processed at each workstation. [6]
- Arborescent Shop: The *arborescent shop* is an m-stage production system, in which each stage has at least one immediate successor except for the last stage (i.e. the finished goods stage), and has only one immediate predecessor except for the first stage (i.e. the raw materials or purchased parts stage). [7]

Sublot Type

- Fixed Sublot: *A fixed sublot* means all products have identical number of items on all stages. [8]
- Equal Sublots: *Equal sublots* means that sublot sizes of each product are fixed.
- Consistent Sublot: A sublot is called *consistent* if it doesn't alter its size over the stages of processing. [9]
- Variable Sublots: In *variable sublot (unequal sublot)* case, the sublot sizes between the stages i and $i+1$ are not equal to those between stages $i+1$ and $i+2$, given the same number of sublots. [9]

Divisibility of the Sublot Size

- Discrete Sublots: For *discrete sublots*, the number of items of a sublot has to be an integer.
- Continuous Sublots: For *continuous* case, no such restriction exists.[8]

Sequence of the Sublots

- Intermingling Sublots: In the multi-product case, if *intermingling sublots* are allowed, the processing of sublots of a product may be interrupted by sublot of other product. In this case, each sublot is treated as an independent product. [8]
- Non-Intermingling Sublots: For *non-intermingling sublots* case, no interruption in the processing of sublots of a product is allowed, which is obviously always given in one-product settings and can be forced in multi-product settings.[8]

Operation Continuity

- No Idling: In *no idling* case, when the sublots start their operation on the same stage, they must finish their operation without interruption.
- Idling: The *idling* case allows idle times.

As known, under the same sublot type, the makespan with idle times generate better results than no idling case. Idling and no-idling cases are illustrated Figures 2.1 and 2.2, respectively. [10]

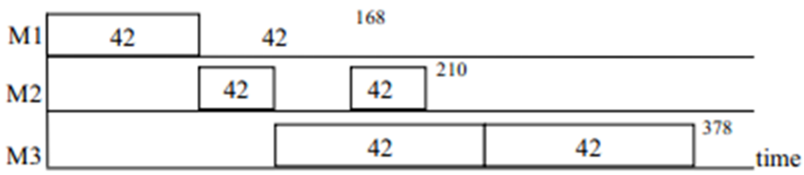


Figure 2.1 Idling Case

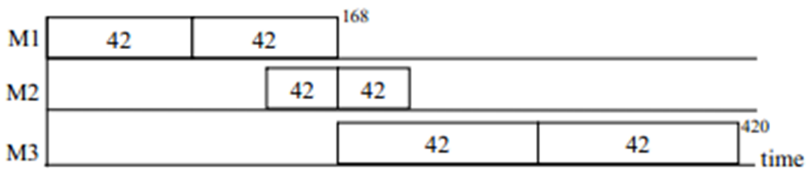


Figure 2.2 No Idling Case

<u>Transfer Timing</u>

- No-wait: In *no-wait schedules*, each sublot has to be transferred to and processed on the next stage immediately after it has been finished on the preceding stage.
- Wait: In a *wait schedule*, sublot may wait for processing between consecutive stages.[8]

<u>Performance Measures</u>

- Time Models: As shown in Table 2-1, the performance of a model depends on minimizing makespan, mean flow time, total flow time, mean tardiness, number of tardy jobs and total deviation from due date.
- Cost Models: The performance of a model depends on minimizing the total cost.

Activities Involved

- Setup: If *attached setups* are required the setup cannot start until the sublot is available at the particular stage. In a *detached setup* the setup is independent from the availability of the sublot. Sometimes setup times are neglected or do not occur.
- Production: Even for the time model, production time is important; for the cost model the inventory type must be considered.
- Transportation: Transportation activity includes the movement of a sublot between stages and the return of an empty transponder. For cost models, the transportation cost per trip is the only important component. For time models, the load and unload times, transportation time, return time of transporter, and the number of capacitated transporters should be considered. Sublot size dependent transfer times can also be considered. Note that the extent to which the transportation activity affects the makespan depends on the number of capacitated transporters [5].

2.2 Evolution of Lot Streaming

Lot streaming problem is originally identified by Reiter [11] in 1966 and rediscovered in the late 1980s to early 1990s. If we get back in the history, in 1964, as a response to the Toyota Manufacturing Program, Joseph Orlicky developed material requirements planning (MRP) [12]. MRP serves as a center organizer that translates the overall production plan into a series of specific steps for achieving the planned production. But MRP has the following weaknesses:

- MRP system is not able to get rid of the uncertainties of production parameters. It assumes that production parameters such as lot sizes and lead times could be determined a priori, external to the system and kept fixed.

- It ignores the finite capacity constraints and focus on material flow.
- All the operations of a lot are processed on a machine before transferring the lot to the other machine.

To get rid of these disadvantages, as an extension of MRP, manufacturing resource planning (MRPII) is developed. MRP II is an integrated method of operational and financial planning for manufacturing companies. Hence, MRPII [13] systems provide better control of inventories and quality improved scheduling, quality control and design control, reduction of working capital of inventory.

By the time, in the 1980s, just-in-time (JIT) manufacturing approach and optimized production technology (OPT) are appeared. JIT is a methodology aimed primarily at reducing flow times within production as well as response times from suppliers and to customers [14]. However, in spite of allowing overlapping of operations, JIT fails to optimality of using unit-size sublots given that these might be suboptimal in a majority of production environments where significant amounts of transfer times and setup times are incurred. OPT aims to reduce the waste in manufacturing system when paying more attention to critical resources than JIT does. OPT uses large process batches to eliminate setup costs and small transfer batches to reduce inventory carrying costs. So it maximizes throughput while eliminating the overall cost. But long setup times of machines, process variability and unbalanced workload cast a suspicion on the success of OPT.

At the end of 1980s, a new technique called constant work-in-process (CONWIP) was introduced to get rid of the weaknesses of JIT. CONWIP allows the simultaneous processing of different types of lots and it makes CONWIP more flexible than JIT. But, it doesn’t solve the sublot sizing or lot sequencing issues.

2.3 Single-Product Lot Streaming

As described in the previous section, processing through the use of transfer lots on several machines were introduced by JIT and OPT approaches in 1980s. Szendrovits [15] published his study that is one of the first papers that introduces the lot streaming approach for minimizing cost for the single-product, multi-stage lot streaming problem with continuous and equal sublots with no-idling case. Even this study doesn't include transportation activities, many other researches and studies are referenced Szendrovits's work. In 1976, Goyal [16] extended Szendrovits's work by developing an algorithm to obtain the optimal sublot sizes. He added transportation cost to Szendrovits's work and created a new algorithm to determine the production lot size and number of sublots for the single-product multi-stage productions. Again in the same year, Szendrovits extended Goyal's study and present a simpler and faster model to minimize the total cost.

Truscott [17] introduced a model for the single-product, multi-stage lot streaming problem with variable sublot in 1986. This model includes setup, operations and load movements between operations. The first objective of this model is to minimize the total production time for the lot. The second aim is to minimize the number of load movements. This approach develops a branch-and-bound algorithm to solve sub problems of scheduling transportation activities. But solving sub problems as zero-one mixed integer programs makes this approach too complex especially for small problems.

In 1989, Potts and Baker [18] created a model for the single-product lot streaming problem up to three machines for minimizing makespan when lot streaming is invoked. They show that they can always find the optimal scheduling policy with consistent sublots when the number of the machines are less than or equal to three.

In 1990, Kropp and Smunt [19] released a paper for the single-product, multi-stage lot streaming problem with equal and consistent sublots. The main propose of the algorithm to minimize the makespan or mean flow time. The makespan problem was modeled as linear programming model while the mean flow time model as a quadratic programming model. They determined the optimal way of splitting a job into sublots under various setup times to run time ratios, number of machines in the flow shop, and number of allowed sublots by using quadratic programming approach to the mean flow time problem. At the same year, Baker and Pyke [20] presented a model for single-product multi-machine flow shops. They used only two sublots to minimize cycle time. Later on, their study was used as base to create the concept of *bottleneck machine*.

In 1993, Trietsch and Baker [10] studied the single-product two-machine flowshop problem for continuous and discrete sublots. They created a model for more than one transporter to minimize the makespan. Also the same year Baker and Jia [21] were created a model for single-product lot streaming problems for product lines with three machines. They researched effects of different constraints, i.e. no idling time, using of equal and consistent sublots, on the makespan value.

Glass, Gupta and Potts [22] developed an algorithm to minimize the makespan for a single job in three-stage production processes. They considered the continuous and consistent sublots on each machine. This algorithm characterized a critical path structure for optimal solution and showed that for the open shop, to minimize makespan, constant time is required. In 1998, Chen and Steiner [23] extended the study of Glass et al. with no setup time to the case of attached setups in a multi-machine flow shops. They showed that no-wait schedules are more convenient in some specific conditions. Again in I998, Sen, Topaloglu and Benli [24] studied single-product lot streaming problems with equal, consistent and variable sublots. Their study justified that equal sublots give more effective results in their conditions.

In 1999, Sriskandarajah and Wagneur [25] considered the problem of minimizing makespan in two-machine no-wait flowshops with multiple products requiring lot streaming. They considered the number of sublots for each product was fixed. They reached the solution that when the flowshop produces only a single product; they obtained optimal continuous-sized sublots. It means that these sublot sizes were also optimal for the problem of simultaneous lot streaming and scheduling of multiple products.

In 2000, Kumar, Bagchi and Sriskandarajah [26] extended the heuristic of Sriskandarajah and Wagneur for the multiple machine case. They showed that, using linear programming approach for one type product, usage of continuous sized sublots gives optimal result. Again at the same year, Ramasesh et al. [27] presented an economic production lot size model for the single item multi-stage manufacturing system with equal sublots and no idling case using lot streaming. This heuristic minimizes the total relevant cost including the cost of setup, transportation and finished goods.

In 2001, Kalir and Sarin [28] developed a heuristic for single-product flowshop manufacturing systems to split a lot into sublots to optimize different performance measures especially the objective function of makespan. In the same year, Bogaschewsky et al. [29] presented a deterministic model for single-product multi-stage lot streaming problem including transportation activities and cost objective. For equal sublots, they generated an algorithm to find optimal number of sublots. For variable sublots, they suggested two algorithms, one is an heuristic, ant the other is an optimal seeking.

In 2003, Kalir and Sarin also released an optimal solution algorithm for the single-batch problem with sublot attached setups [30]. This heuristic guarantees the near-optimal solution in a fast and efficient way. In the same year Chen and Steiner [31] showed that the addition of the no-wait constraint in a regular flowshop doesn't affect the minimum makespan for the single-product lot streaming problem in no-wait flowshops. Van Nieuwenhuyse and Vandaele [32] created a cost minimization model for a single-product deterministic flowshop lot streaming problem. In their approach, they assumed that sublot sizes are discrete and equal-sized to minimize the sum of inventory holding, transportation costs and gap costs. As a result, they reached that adding gap cost to the total cost function gives the same results as a no lot splitting case.

In 2004, Chiu et al. [33] developed a binary mixed integer programming for a single-product, multi-stage lot streaming problem to minimize total cost including the transportation and makespan costs. They proposed two heuristics. The first one extended the two-stage method of Trietsch and Baker (1993). The second heuristic was built to relax the transporter capacity constraints.

In 2005, Chiu and Chang [7] released two models for a multi-stage flowshop lot streaming problems. In their models, the sublot sizes are assumed to be equal, the number of transporters and the capacity of them are assumed to be infinite. They carried out an experimental design for the cost factors and analyze a number of different levels.

2.4 Multi-Product Lot Streaming

In the literature, studies mainly focuses on simple lot streaming problems. Because when the scale of problem is expanded, its complexity increases. To get rid of this complexity, researchers partition the multi-product lot streaming problem into a sub problem, propose different heuristic approaches to them and solve these partitions individually. In this section, we explain the heuristic approaches of different kinds of multi-product lot streaming problems' solutions.

In 1985, Truscott was first studied single job, equal sized sublots lot streaming problem on multiple machines by considering the setup times [34]. In 1992, Vickson and Alfredsson modified the Johnson's rule to obtain optimal solution for unit sized and equal sized sublots by ignoring transfer and setup times for multiple jobs on two and three machines flowshops system [35]. They created the *non-intermingled* solution which is among many optimal solutions there exists an optimal solution where sublots of the same products are processed continuously on each machine [4]. Again in 1992, Cetinkaya and Kayaligil extended the study of Vickson and Alfredsson by considering detached setups [36]. Their heuristic was very similar to Johnson's rule and showed that splitting jobs into sublots and sequencing them could be done independently. In 1994, Cetinkaya studied on lot scheduling problem to minimize the maximum makespan for two-stage flow shops in which the movement of transfer batches (sublots) from the first stage to the next were allowed when set-up, processing and removal times were considered as separable and independent of the order in which jobs were processed at any of two stages [37]. In 2001, Kalir and Sarin released *a bottleneck minimal idleness heuristic*, for the multi-product lot streaming problem [28]. This heuristic minimizes the idle time between sublots of each product and gives very close solution to optimal. Again in 2001, Kalir and Sarin extended their heuristic for multiple jobs by excluding setup times. In 2009, Laha and Sarin [38] and in 2011, Glass and Possani [39] referred to this heuristic in their studies.

In 1993, Trietsch and Baker presented linear and integer programming formulation for a single job using continuous and discrete values of consistent sublots on a three machine flowshop systems [10]. In 2001, Wagneur added the no-wait condition to Trietsch and Baker's formula [40]. In 1997 and 1998, Chen and Steiner extended this case by detached [41] and attached setup times [23]. In 2000, Kumar, Bagchi and Sriskandarajah extended the two-staged approach of Sriskandarajah et al.[25] for the case of multi-product, multi-stage, no-wait flowshop environment with non-intermingled and discrete sublots using three-staged approach [26]. In 2002, Buckhin, Tzur and Jaffe presented *single machine bottleneck* procedure [42] which guarantees very close solution to optimal solution and optimal solution for some special cases for two-machine, sublot-attached flowshop lot streaming problems. In 2003, Hall et al. studied on Sriskandarajah and Wagneur's problem [25] by attaching setup times to it [43]. They reached an efficient solution for the multi-stage, no-wait multi-product lot streaming problem with consistent non-intermingled integer sublot sizes. In 2005, they modified their heuristic for no-wait two-machine open shops with consistent non-intermingled sublots [43]. With this heuristic, they reached good results for two-machine flowshops with up to 50 products.

In 2002, Yoon and Venture developed a linear programming for no-wait lot streaming flowshops to find the optimal sequence that minimize the absolute deviation [44]. In order to accelerate production, a job was allowed to overlap its operations between successive machines and by splitting it into a number of smaller sublots and moving the completed potion of the sublots to downstream machines[45]. They also developed a hybrid generic algorithm for buffers between successive machines having infinite capacities and sublots are equal sized and buffers between successive machines having finite capacities and sublots are consistent [46].

In 2004, Hug, Cutright and Martin developed an integer programming model to obtain optimum sublot sizes while enumerating the number of sublots for multi-product lot streaming problem using discrete sublots [47]. In 2007, 2008 and 2009 Marimuthu, Ponnambalam and Jawahar released a *tabu search, a simulated annealing, hybrid generic algorithm, ant colony optimization* and *threshold accepting* algorithms which include setup times [48], [49, 50].

In 2005, Zhang, Yin, Liu and Linn proposed two heuristics to minimize the mean completion time for multi-job lot streaming problem in two-stage hybrid flowshops with *m* identical machines at the first stage and a single machine at the second stage [51].

In 2009, Martin presented a hybrid genetic algorithm/mathematical programming heuristic for the *n*-job, *m*-machine flowshop problem with lot streaming. The number of sublots for each job and the size of sublots were directly addressed by the heuristic and setups may be sequence-dependent. A new aspect of this problem, the interleaving of sublots from different jobs in the processing sequence, were developed and addressed [52].

In 2011, Buscher and Shen proposed an integer programming formulation to solve multi-product lot streaming problem in a job shop environment where setup times are involved. They optimally solved this problem for consistent sublots [53]. F.M. Defersha and M. Chen developed a hybrid genetic algorithm for a model that appeared in recent literature for one -job m -machine lot streaming problems with variable sublots and setup and showed that the performance of the proposed genetic algorithm is encouraging in the same year [54].

In 2012, M. Karimi and Nasab presented a mathematical modeling of joint lot sizing and scheduling problem in job shop environment under a set of working conditions. They deal with process compressibility and their further experiences on random test data confirmed that the performance of the proposed method with less than 5.02% optimality gap while solving the problems in very shorter times than CPLEX [55].

In 2013, N. Mortezaei and N. Zulkifli developed a mathematical model for the integration of lot sizing and flow shop scheduling with lot streaming. They developed a mixed-integer linear model for multiple products lot sizing and lot streaming which enabled the operator to find optimal production quantities, optimal inventory levels, optimal sublot sizes, and optimal sequence simultaneously [56]. With this research they showed that the best makespan shall be achieved through the consistent sublots with intermingling case.

.

CHAPTER 3

PROBLEM DEFINITION AND MATHEMATICAL MODELS

In this chapter we first define our problem under consideration for both equal and unequal sublots cases and then propose mathematical models for solving these problems.

3.1 Problem Statement

There is a set of N jobs (product lots) to be processed on a two-machine flowshop in which both machines $M1$ and $M2$ operate independently and ready at time zero for processing jobs. All jobs are available at time zero and processed first on $M1$ and then on $M2$. That is, the first and second operations of the jobs are performed by machines $M1$ and $M2$, respectively. The setup times required before processing each job and the transfer time from machine $M1$ to machine $M2$ are assumed to be zero, and ignored.

In our study, we assume that the total number of allowed sublots for all jobs is S (where $S > N$) and known in advance. Moreover, only one job can be processed on a machine at a time and preemption is not allowed, i.e. the processing of any sublot cannot be interrupted on any machine at any time and resumed at a later time.

Our problem is to determine the number of sublots for each job, the size of the each sublot and the processing sequence of all sublots that gives the minimum makespan. We investigate the problem for two cases: unequal sized sublots and equal sized sublots. Details of these approaches will be explained at the following sections.

3.2 Mathematical Model for the Case with Unequal Sized Sublots

Based on the problem characteristics and assumptions given in Section 3.1, we developed a mixed integer linear programming (MILP) model for solving optimally the lot streaming problem with unequal sized sublots. This model aims to determine the sublot sizes for each product and the processing sequence of the sublots of all products for the case with unequal sized sublots. Below, we present parameters, indices and variables are used in this model.

Parameters and Indices:

N Number of jobs

S Total number of sublots allowed for all jobs (where $S > N$)

j Index for jobs ($j = 1,2,..., N$)

t Position index for sublots in the sequence ($t = 1,2,..., S$)

m Index for machines ($m = 1,2$)

Q_j Lot size of job j

$p_{j,m}$ Unit processing time of job j on machine m

L Sufficiently large positive number

Decision Variables:

$$Z_{j,t} = \begin{cases} 1 & \text{if} \quad \text{a sublot of job } j \text{ is assigned to position } t \text{ in the sequence} \\ 0 & \text{otherwise} \end{cases}$$

$X_{j,t}$ = Size of the sublot which belongs to job j assigned to position t

$C_{t,m}$ = Completion time of the sublot assigned to position t on machine m

$C_{\max}$ = Makespan (completion time of all jobs on Machine 2)

MILP:

Minimize:

$$C_{\max} = C_{S,2} \tag{1}$$

Subject to:

$$\sum_{j=1}^{N} Z_{j,t} = 1 \qquad \text{for } t = 1,2,...,S \tag{2}$$

$$X_{j,t} \le L \cdot Z_{j,t} \qquad \text{for } j = 1,2,...,N;\ t = 1,2,...,S \tag{3}$$

$$\sum_{t=1}^{S} X_{j,t} = Q_j \qquad \text{for } j = 1,2,...,N \tag{4}$$

$$C_{0,m} = 0 \qquad \text{for } m = 1,2 \tag{5}$$

$$C_{t,0} = 0 \qquad \text{for } t = 1,2,...,S \tag{6}$$

$$C_{t,m} \ge C_{t-1,m} + \sum_{j=1}^{N} p_{j,m} \cdot X_{j,t} \qquad \text{for } t = 1,2,...,S;\ m = 1,2 \tag{7}$$

$$C_{t,m} \ge C_{t,m-1} + \sum_{j=1}^{N} p_{j,m} \cdot X_{j,t} \qquad \text{for } t = 1,2,...,S;\ m = 1,2 \tag{8}$$

$$Z_{j,t} \in \{0,1\} \qquad \text{for } \forall j,t \tag{9}$$

$$X_{j,t} \ge 0 \qquad \text{for } \forall j,t$$

$$C_{t,m} \ge 0 \qquad \text{for } \forall t,m$$

In the above presented MILP model, the objective function $C_{\max}$ in (1) is to minimize the makespan, which is the completion time of the last sublot in the processing sequence of the products. The Constraint Set (2) ensures that each position in the sequence is occupied by a sublot of a job. The Constraint Set (3) guarantees that size of a sublot becomes positive if this sublot is assigned to a position in the sequence. Constraint Set (4) ensures that the sum of all sublot sizes of a job equals to the lot size of this job. Constraint Sets (5) and (6) are initialization constraints for completion times of sublots on each machine. Constraint Set (7) guarantees that the completion time of a sublot on a machine should be greater than or equal to the sum of completion time of the sublot in the previous position and the processing time of this sublot on the same machine. Constraint Set (8) ensures that the completion time of a sublot on a machine should be greater than or equal to the sum of the completion time of this sublot on the previous machine and the processing time of this sublot on the current machine. Constraint Set (9) imposes binary and non-negativity restrictions on the decision variables, respectively.

In this MILP model, two sets of the decision variables are continuous variables, and the number of this type of decision variables is $S \times (N+2)$. However, there is only one set of decision variables, which has $S \times N$ binary. This means that there are totally $2S \times (N+1)$ decision variables. On the other hand, the MILP model has $6S + N \times (S+1) + 2$ constraints.

3.3 Mathematical Model for the Case with Equal Sized Sublots

The second model aims to determine sublot sizes for each product and the processing sequence of the sublots of all products for the case with equal sized sublots. Additional parameters, indices and variables for our model to solve the case with equal sized sublots are given below:

Additional Parameters and Indices:

$S - N + 1$ Maximum number of sublots allowed for a job

k Index for sublots ($k = 1,2,...,S - N + 1$)

Additional Decision Variables:

$$Y_{j,k} = \begin{cases} 1 & \textit{if} \quad \text{job } j \text{ is split into } k \text{ sublots} \\ 0 & \textit{otherwise} \end{cases}$$

$X_{j,k,t}$ Non-negative continuous variable

MILP:

Minimize:

$$C_{\max} = C_{S,2} \tag{10}$$

Subject to:

$$\sum_{k=1}^{S-N+1} Y_{j,k} = 1 \qquad \text{for } j = 1,2,\ldots,N \tag{11}$$

$$\sum_{j=1}^{N} \sum_{k=1}^{S-N+1} k \cdot Y_{j,k} = S \tag{12}$$

$$\sum_{j=1}^{N} Z_{j,t} = 1 \qquad \text{for } t = 1,2,\ldots,S \tag{13}$$

$$\sum_{t=1}^{S} Z_{j,t} = \sum_{k=1}^{S-N+1} k \cdot Y_{j,k} \qquad \text{for } j = 1,2,\ldots,N;\ k = 1,2,\ldots,S-N+1 \tag{14}$$

$$X_{j,k,t} \geq Y_{j,k} + Z_{j,t} - 1 \qquad \text{for } j = 1,2,\ldots,N;\ k = 1,2,\ldots,S-N+1;\ t = 1,2,\ldots,S \tag{15}$$

$$C_{0,m} = 0 \qquad \text{for } m = 1,2 \tag{16}$$

$$C_{t,0} = 0 \qquad \text{for } t = 1,2,\ldots,S \tag{17}$$

$$C_{t,m} \geq C_{t-1,m} + \sum_{j=1}^{N} \sum_{k=1}^{S-N+1} \frac{Q_j}{k} \cdot p_{j,m} \cdot X_{j,k,t} \qquad \text{for } t = 1,2,\ldots,S;\ m = 1,2 \tag{18}$$

$$C_{t,m} \geq C_{t,m-1} + \sum_{j=1}^{N} \sum_{k=1}^{S-N+1} \frac{Q_j}{k} \cdot p_{j,m} \cdot X_{j,k,t} \qquad \text{for } t = 1,2,\ldots,S;\ m = 1,2 \tag{19}$$

$$Y_{j,k} \in \{0,1\} \qquad \text{for } \forall j,k \tag{20}$$

$Z_{j,t} \in \{0,1\}$ for $\forall j,t$

$X_{j,k,t} \geq 0$ for $\forall j,k,t$

$C_{t,m} \geq 0$ for $\forall t,m$

In the above MILP model, the objective function $C_{\max}$ in (10) is to minimize the makespan, which is the completion time of the last sublot in the sequence. Constraint Set (11) ensures that a job is split into at most $S-N+1$ sublots. Constraint Set (12) guarantees that the sum of sublots of all jobs is equal to the total number of sublots for all jobs. Constraint Set (13) ensures that each position in the sequence is occupied by a sublot of a job. Constraint Set (14) guarantees that total number of positions occupied by a job is equal to the total number of sublots of this job. Constraint Set (15) determines the values of the continuous variables $X_{j,k,t}$'s. Constraint Set (16) and (17) are initialization constraints for completion times. Constraint Set (18) guarantees that the completion time of a sublot on a machine should be greater than or equal to the completion time of the sublot in the previous position plus the processing time of this sublot on the same machine. Constraint Set (19) ensures that the completion time of a sublot on a machine should be greater than or equal to the completion time of this sublot on the previous machine plus the processing time of this sublot on the current machine. Constraint Set (20) imposes binary and non-negativity restrictions on the decision variables, respectively.

In this MILP model, two sets of the decision variables are continuous variables, and the number of this type of decision variables is $S \times (N+2)$. However, there are two sets of binary decision variables, which have totally $N \times (2S-N+1)$ binary variables. This means that there are totally $2S \times (N+1)$ decision variables. On the other hand, the MILP model has $Nx[S(S-N+2)-N+2]+6S+3$ constraints.

CHAPTER 4

HEURISTIC ALGORITHMS

The size of the the MILP models increases drastically with the increase in the number of jobs and the total number of sublots allowed. Therefore, the optimal solutions to the large-scale problems are not likely to be obtained within reasonable computational times. Moreover, the existence of a polynomial-time algorithm to solve the problem optimally is unlikely since we have an NP-hard problem. This motivated us to develop fast algorithms that provide near-optimal solutions.

In this chapter we present our proposed heuristic algorithms for solving the two cases, unequal sized and equal sized sublots, of the lot streaming problem under consideration.

4.1 Heuristic Algorithm for the Case with Unequal Sized Sublots

By this heuristic, we aim to determine the number of the sublots for each product, the size of each sublot on each machine and the processing sequence of the sublots for minimizing the makespan for multi-product lot streaming problem with unequal sublots. To address this problem, we extend the heuristic algorithm of Cetinkaya [37]. Below, we present the notation and formulation of our heuristic for the multi-product flowshop lot streaming problem with unequal sublots:

Parameters:

M	Number of machines ($M = 2$)
N	Number of jobs
S	Total number of sublots
Q_n	Lot size of job n
$P_{n,m}$	Processing time for one unit of job n on machine m
$TP_{n,m}$	Total processing time for one unit of job n on machine m
K_n	Number of sublots of job n
$x_{s,n}$	Size of the sublot s of job n
$k_{s,n}$	Total process time for sublot s of job n
Z_n	Idle time for job n
$f_{s,n}$	Fraction factor for sublot s of job n
$TSP_{n,,s,m}$	Total processing time for sublot s of job n on machine m

Indices:

m	Machine index where $m = 1,...,M$
n	Job index where $n = 1,...,N$
s	Sublot index where $s = 1,...,K_n$

Heuristic Algorithm for Unequal Sized Sublots Case:

Step 1:

Identify the maximum process time on M_1 and M_2, that is:

$$\exists n, m\,,\ max\left\{TP_{n,m}\right\} \text{ where } TP_{n,m} = P_{n,m}.Q_n$$

The job with the highest $TP_{n,m}$ value will be the primary job to calculate sublot sizes.

Step 2:

Identify number of sublot used by job with the highest $TP_{n,m}$ value ($JHTP_{n,m}$). We assume that each job except $JHTP_{n,m}$ value has only one sublot and the rest of the sublots belong to $JHTP_{n,m}$. That is,

$$K_n = \begin{cases} 1 \text{ if job } n \text{ is not } JHTP_{n,m} \\ S-(N-1) \quad \text{otherwise} \end{cases}$$

Step 3:

Identify the size of each sublot of $JHTP_{n,m}$ using the following algorithm:

$$x_{s,n} = \begin{cases} \left(\dfrac{\alpha_n - 1}{\alpha_n^{K_n} - 1}\right) . Q_n \\ \alpha_n^{s-1} . x_1 \quad , \forall s \;\; where\; 1 < s \le K_n \end{cases}$$

where

$$\alpha_n = \frac{P_{n,2}}{P_{n,1}} \text{ and } n \text{ is the index of } JHTP_{n,m}.$$

The jobs except $JHTP_{n,m}$ aren't split. For these jobs sublot size equals to lot size, i.e.

$$x_{1,n} = Q_n \text{ where } n \text{ is NOT the } JHTP_{n,m}.$$

Step 4:

If, at least, one of the sublot size calculated at Step 3, is non-integer sized go to Step 4.1 to recalculate the integer sized ones. Otherwise, go to Step 5.

Step 4.1 (Converting non-integer sized sublot to integer-sized sublot)

For all jobs,

Step 4.1.1 :

Calculate the idle time Z_n as

$$Z_n = \max_{1 \le u \le K_n} \left\{ \sum_{s=1}^{u} P_{n,1}.x_{s,n} - \sum_{s=1}^{u-1} P_{n,2}.x_{s,n} \right\} \text{ where } n \in [1, N]$$

Step 4.1.2 :

For each sublot *s,* calculate the integer sized sublot

$$x_{s,n} = \min \left\{ \left\lfloor \left[Z_n - \left(P_{n,1} - P_{n,2}\right) \sum_{m=1}^{s-1} x_{m,n} \right] \Big/ P_{n,1} \right\rfloor, Q_n - \sum_{m=1}^{s-1} x_{m,n} \right\}$$

Step 4.1.3 :

If $\sum_{i=1}^{K_n} x_{i,n} = Q_n$, then the calculated sublot sizes, that are $x_{s,n}$, are not integer sublot sizes go to Step 2 using following Z_n.

$$Z_n = Z_n + P_{n,1} \min \left(1 - f_{s,n}\right)$$

where

$$f_{s,n} = \left[Z_n - \left(P_{n,1} - P_{n,2}\right) \sum_{m=1}^{s-1} x_{m,n} \right] \Big/ P_{n,1} - x_{s,n} \ .$$

Otherwise,

1. If there are no zero sized sublots, go to Step 5.
2. If there exist zero sized sublots, transfer them to the next job with greatest $TP_{n,m}$ after the current job.

Step 5:

Calculate the total process time of each sublot ($TSP_{n,,s,m}$) on both machine and group them as *Set 1* and *Set 2.*

Set 1 is a set of sublots that are processed on M_1 at most in the time that are processed on M_2. In other words; each sublot of *Set* 1 is processed on M_1 in less or equal time on M_2.

Set 2 is a set of sublots that are processed on M_1 longer than that are processed on M_2.

Set 1 and *Set 2* are mathematically expressed as

$$Set\ 1 = \{ k_n : TSP_{n,s,1} \leq TSP_{n,s,2} \}$$

$$Set\ 2 = \{ k_n : TSP_{n,s,1} > TSP_{n,s,2} \}$$

where $TSP_{n,,s,m} = x_{s,n.} P_{n,m}$, $n \in [1, N]$, $k_n \in [1, K_n]$ and $m = \{1,2\}$.

Step 6:

Optimize *Set 1* and *Set 2* by rearranging their entities. While *Set 1* is optimized by sorting sublots according to their increasing process time on M_1 ; *Set 2* is optimized by sorting sublots according to their decreasing process time on M_2. If we call optimized *Set 1* as *OSet 1* and optimized *Set 2* as *OSet2,* the mathematical representation of *OSet 1* and *OSet 2* are as follows:

$$OSet\ 1 = \{ \forall k_n \in Set\ 1 : TSP_{n,s,1} \leq TSP_{n+1,s,1} \}$$

$$OSet\ 2 = \{ \forall k_n \in Set\ 2 : TSP_{n,s,2} \geq TSP_{n+1,s,2} \}$$ where $n \in [1, N]$, $k_n \in [1, K_n]$.

Step 7:

To minimize the makespan, sublot sequence is needed to be optimized. Optimized sublot sequence, *OS,* is a combination of *Set 1* and *Set 2* as all elements of *Set 1* is followed by all elements of *Set 2*. *OS* is represented mathematically as follows:

$$OS = \bigcup_{n=1}^{S_1} k_n \cup \bigcup_{m=1}^{S_2} k_m$$

where, $k_n \in Set\,1$, $k_m \in Set\,2$, S_1 and S_2 are the sizes of *Set 1* and *Set 2*, respectively.

4.1.1 Numerical Example

In this section, we provide a numerical example for illustrating the heuristic algorithm for solving the unequal sized sublot case. Consider a simple instance of the problem in which there are 5 jobs and the total number of sublots is 20. Unit processing times on the machines and the lot sizes for all jobs are given in Table 4-1.

Table 4-1 Process Times on M1 and M2

JOB NO	Process Time on M1	Process Time on M2	Total Lot Size
Job 1	3	6	12
Job 2	1	7	17
Job 3	1	4	16
Job 4	1	9	20
Job 5	4	6	5

First we will find the job with the maximum process time. As seen on Table 4-1, the job with the maximum process time is Job 4. So, Job 4 will have the maximum number of sublots when other jobs have only one sublot, which is equal to the lot. It means that

$$K_4 = 20 - 4 = 16 \text{ and } K_1 = K_2 = K_3 = K_5 = 1.$$

Now, we shall find the sizes of sublots for each job. Here it is obvious that for Jobs 1, 2, 3 and 5, the sublot equals to the total lot size of the job since these jobs are not split into sublots and have only one sublot. But Job 4 is split in to 16 sublots. Our sublot factor α_4 is

$$\alpha_4 = \frac{P_{4,2}}{P_{4,1}} = \frac{9}{1} = 9.$$

When we calculate the size of each sublot in Job 4 using α_4, we obtain:

$$x_{1,4} = \left(\frac{9-1}{9^{16}-1}\right).20 = 8.63x10^{-14}$$

$$x_{2,4} = \alpha.x_{1,4} = 7.77x10^{-13}$$

$$x_{3,4} = \alpha^2.x_{1,4} = 6.99x10^{-12}$$

$$x_{4,4} = \alpha^3.x_{1,4} = 6.29x10^{-11}$$

$$x_{5,4} = \alpha^4.x_{1,4} = 5.66x10^{-10}$$

$$x_{6,4} = \alpha^5.x_{1,4} = 5.09x10^{-9}$$

$$x_{7,4} = \alpha^6.x_{1,4} = 4.58x10^{-8}$$

$$x_{8,4} = \alpha^7.x_{1,4} = 4.12x10^{-7}$$

$$x_{9,4} = \alpha^8.x_{1,4} = 3.71x10^{-6}$$

$$x_{10,4} = \alpha^9.x_{1,4} = 3.34x10^{-5}$$

$$x_{11,4} = \alpha^{10}.x_{1,4} = 3.01x10^{-4}$$

$$x_{12,4} = \alpha^{11}.x_{1,4} = 2.70x10^{-3}$$

$$x_{13,4} = \alpha^{12}.x_{1,4} = 0.02$$

$$x_{14,4} = \alpha^{13}.x_{1,4} = 0.21$$

$$x_{15,4} = \alpha^{12}.x_{1,4} = 1.97$$

$$x_{16,4} = \alpha^{12}.x_{1,4} = 17.7$$

After calculating the size of each sublot, we shall categorize these sublots as Set 1 and Set 2. As shown in Table 4-2, each sublot's process time on M1 is less than its process time on M2. Thus, it is obvious that all sublots belong to Set 1.

$$Set\ 1 = \left\{ \begin{array}{l} k_{1,1}, k_{1,2}, k_{1,3}, k_{1,5}, k_{1,4}, k_{2,4}, k_{3,4}, k_{4,4}, \\ k_{5,4}, k_{6,4}, k_{7,4}, k_{8,4}, k_{9,4}, k_{10,4}, k_{11,4}, k_{12,4}, k_{13,4}, k_{14,4}, k_{15,4}, k_{16,4} \end{array} \right\}$$

$Set\ 2$ is empty.

Now, to find the sequence of sublots that provides the optimal makespan, we will find $OSet\ 1$ and $OSet\ 2$. While $Set\ 2$ is empty, $OSet\ 2$ is empty too. But to obtain $OSet\ 1$, we rearrange $Set\ 1$ as

$$OSet\,1 = \left\{ \begin{matrix} k_{1,4}, k_{2,4}, k_{3,4}, k_{4,4}, k_{5,4}, k_{6,4}, k_{7,4}, k_{8,4}, k_{9,4}, k_{10,4}, \\ k_{11,4}, k_{12,4}, k_{13,4}, k_{14,4}, k_{15,4}, k_{1,3}, k_{1,2}, k_{16,4}, k_{1,5}, k_{1,1} \end{matrix} \right\}.$$

Table 4-2 Total Sublot Process Times of Jobs on M1 and M2

SUBLOT NO	Total Sublot Process Time on M1	Total Sublot Process Time on M2
$x_{1,1}$	12x3=36	12x6=72
$x_{1,2}$	17x1=17	17x7=119
$x_{1,3}$	16x1=16	16x4=64
$x_{1,5}$	5x4=20	5x6=30
$x_{1,4}$	$8.63x10^{-14}x1= 8.63x10^{-14}$	$8.63x10^{-14}x9 = 7.77x10^{-13}$
$x_{2,4}$	$7.77x10^{-13}x1 = 7.77x10^{-13}$	$7.77x10^{-13}x9= 6.99.10^{-12}$
$x_{3,4}$	$6.99x10^{-12}x1 = 6.99x10^{-12}$	$6.99x10^{-12}x9 = 6.29.10^{-11}$
$x_{4,4}$	$6.29x10^{-11}x1 = 6.29x10^{-11}$	$6.29x10^{-11}x9 = 5.66x10^{-10}$
$x_{5,4}$	$5.66x10^{-10}x1 = 5.66x10^{-10}$	$5.66x10^{-10}x9 = 5.09x10^{-9}$
$x_{6,4}$	$5.09x10^{-9}x1 = 5.09x10^{-9}$	$5.09x10^{-9}x9 = 4.58x10^{-8}$
$x_{7,4}$	$4.58x10^{-8}x1 = 4.58x10^{-8}$	$4.58x10^{-8}x9 = 4.12x10^{-7}$
$x_{8,4}$	$4.12x10^{-7}x1 = 4.12x10^{-7}$	$4.12x10^{-7}x9 = 3.71x10^{-6}$
$x_{9,4}$	$3.71x10^{-6}x1 = 3.71x10^{-6}$	$3.71x10^{-6}x9 = 3.34x10^{-5}$
$x_{10,4}$	$3.34x10^{-5}x1 = 3.34x10^{-5}$	$3.34x10^{-5}x9 = 3.01x10^{-4}$
$x_{11,4}$	$3.01x10^{-4}x1 = 3.01x10^{-4}$	$3.01x10^{-4}x9 = 2.70x10^{-3}$
$x_{12,4}$	$2.70x10^{-3}x102.70x10^{-3}$	$2.70x10^{-3}x9 = 0.02$
$x_{13,4}$	0.02x1=0.02	0.02x9=0.21
$x_{14,4}$	0.21x1=0.21	0.21x9=1.97
$x_{15,4}$	1.97x1=1.97	1.97x9=17.7
$x_{16,4}$	17.7x1=17.7	17.7x9=159.3

As seen on Table 4-2, almost all sublots are non-integer sized. So we need to recalculate integer sized sublots. To do this, first we calculate the idle time Z_n where $n \in [1, N]$. Our first idle time is $8.63x10^{14}$. Now by using this idle time we calculate non-integer sized sublots of Job 4 using the following formula.

$$x_{s,n} = \min\left\{\left\lfloor \left[Z_n - (P_{n,1} - P_{n,2})\sum_{m=1}^{s-1} x_{m,n} \right] \Big/ P_{n,1} \right\rfloor, Q_n - \sum_{m=1}^{s-1} x_{m,n} \right\}$$

Now, when we recalculate integer sized sublots, we will see all of them are zero. That is,

$$x_{1,4} = \min\left\{\left\lfloor \left[8.63x10^{-14} - (1-9).0\right] \big/ 1 \right\rfloor, 20-0 \right\} = \min\left\{\left\lfloor 8.63x10^{-14} \right\rfloor, 20-0\right\} = 0$$

$$x_{2,4} = \min\left\{\left\lfloor \left[8.63x10^{-14} - (1-9).0\right] \big/ 1 \right\rfloor, 20-0 \right\} = \min\left\{\left\lfloor 8.63x10^{-14} \right\rfloor, 20-0\right\} = 0$$

$$x_{3,4} = \min\left\{\left\lfloor \left[8.63x10^{-14} - (1-9).0\right] \big/ 1 \right\rfloor, 20-0 \right\} = \min\left\{\left\lfloor 8.63x10^{-14} \right\rfloor, 20-0\right\} = 0$$

$$x_{3,4} = x_{4,4} = x_{5,4} = x_{6,4} = x_{7,4} = x_{8,4} = x_{9,4} = x_{10,4} = x_{11,4} = x_{12,4} = x_{13,4} = x_{14,4} = x_{15,4} = x_{16,4} = 0$$

.

Here, it is obvious that $\sum_{i=1}^{K_4} x_{i,4} \neq Q_4$ since $\sum_{i=1}^{K_4} x_{i,4} = 0$ and $0 \neq 20$. So we need to use the fraction factor $f_{s,4}$ to recalculate the new idle time Z_4 as

$$f_{1,4} = \left(\left[8.63x10^{-14} - (1-9).0\right] \big/ 1 \right) - 0 = 8.63x10^{-14}$$

$$f_{2,4} = \left(\left[8.63x10^{-14} - (1-9).0\right] \big/ 1 \right) - 0 = 8.63x10^{-14}$$

$$f_{3,4} = f_{4,4} = f_{5,4} = f_{6,4} = f_{7,4} = f_{8,4} = f_{9,4} = f_{10,4} = f_{11,4} = f_{12,4} = f_{13,4} = f_{14,4} = f_{15,4} = f_{16,4} = 8.63x10^{-14}$$

Our new idle time Z_4 is

$$Z_4 = Z_4 + P_{4,1} \min(1 - f_{s,4}) = 8.63x10^{-14} + \left(1 - 8.63x10^{-14}\right) = 1.$$

Now, when we recalculate the new sublot sizes using the new idle time $Z_4 = 1$, we obtain:

$$x_{1,4} = \min\left\{\left\lfloor \left[1-(1-9).0\right] \big/ 1 \right\rfloor, 20-0\right\} = \min\{1,20\} = 1$$

$$x_{2,4} = \min\left\{\left\lfloor \left[1-(1-9).1\right] \big/ 1 \right\rfloor, 20-1\right\} = \min\{9,19\} = 9$$

$$x_{3,4} = \min\left\{\left\lfloor \left[1-(1-9).10\right] \big/ 1 \right\rfloor, 20-10\right\} = \min\{81,10\} = 10$$

$$x_{4,4} = \min\left\{\left\lfloor \left[1-(1-9).20\right] \big/ 1 \right\rfloor, 20-20\right\} = 0$$

$$x_{5,4} = \min\left\{\left\lfloor \left[1-(1-9).20\right] \big/ 1 \right\rfloor, 20-20\right\} = 0$$

$$x_{4,4} = x_{5,4} = x_{6,4} = x_{7,4} = x_{8,4} = x_{9,4} = x_{10,4} = x_{11,4} = x_{12,4} = x_{13,4} = x_{14,4} = x_{15,4} = x_{16,4} = 0$$

Here, again we need to check whether that sum of the sublot sizes are equal to the total lot size. We observe that

$$\sum_{i=1}^{K_4} x_{i,4} = 1 + 9 + 10 = 20 = Q_4 \;.$$

When we analyze the sublot sizes, we see that sublot sizes are zero for 13 sublots. It means that 13 of the sublots out of 16 are useless for Job 4. So they will be transferred to Job 2, which is the job with the second maximum process time. Thus, the total number of sublots for Job 2 becomes 13+1=14. Now, we need to find integer sized sublots for Job 2. First, we need to calculate our sublot factor α_2 which is

$$\alpha_2 = \frac{P_{2,2}}{P_{2,1}} = \frac{7}{1} = 7 \cdot$$

If we calculate the sizes of each sublot for Job 2 using α_2, we obtain:

$$x_{1,2} = \left(\frac{7-1}{7^{14}-1}\right).17 = 1.5x10^{-10}$$

$$x_{2,2} = \alpha.x_{1,2} = 1.05x10^{-9}$$

$$x_{3,4} = \alpha^2.x_{1,2} = 7.36x10^{-9}$$

$$x_{4,2} = \alpha^3.x_{1,2} = 5.15x10^{-8}$$

$$x_{5,2} = \alpha^4.x_{1,2} = 3.61x10^{-7}$$

$$x_{6,2} = \alpha^5.x_{1,2} = 2.52x10^{-6}$$

$$x_{7,2} = \alpha^6.x_{1,2} = 1.76x10^{-5}$$

$$x_{8,2} = \alpha^7.x_{1,2} = 1.23x10^{-4}$$

$$x_{9,2} = \alpha^8.x_{1,2} = 8.66x10^{-4}$$

$$x_{10,2} = \alpha^9.x_{1,2} = 6.06x10^{-3}$$

$$x_{11,2} = \alpha^{10}.x_{1,2} = 0.04$$

$$x_{12,2} = \alpha^{11}.x_{1,2} = 0.29$$

$$x_{13,4} = \alpha^{12}.x_{1,2} = 2.08$$

$$x_{14,2} = \alpha^{13}.x_{1,2} = 14.5$$

But it is obvious that sublot sizes are not integer sized and we need integer sized sublots. To do this, we need to calculate the idle time for Job 2 which is

$$Z_2 = \max_{1\le u\le K_2}\left\{\sum_{s=1}^{u} P_{2,1}.x_{s,2} - \sum_{s=1}^{u-1} P_{2,2}.x_{s,2}\right\} = 1.5x10^{-10}$$

Now by using this idle time Z_2 we calculate non-integer sized sublots of Job 2 using the following formula.

$$x_{s,n} = \min\left\{\left\lfloor \left[Z_n - \left(P_{n,1} - P_{n,2}\right)\sum_{m=1}^{s-1} x_{m,n}\right] \Big/ P_{n,1} \right\rfloor, Q_n - \sum_{m=1}^{s-1} x_{m,n}\right\}$$

Now, when we recalculate integer sized sublots, we will see all of them are zero. That is,

$$x_{1,2} = \min\left\{\left\lfloor \left[1.5x10^{-10} - (-6).0\right] \Big/ 1 \right\rfloor, 17-0\right\} = \min\left\{\lfloor 1.5x10^{-10} \rfloor, 17\right\} = 0$$

$$x_{2,2} = \min\left\{\left\lfloor \left[1.5x10^{-10} - (-6).0\right]/1 \right\rfloor, 17-0\right\} = \min\left\{\left\lfloor 1.5x10^{-10}\right\rfloor, 17\right\} = 0$$

$$x_{3,2} = x_{4,2} = x_{5,2} = x_{6,2} = x_{7,2} = x_{8,2} = x_{9,2} = x_{10,2} = x_{11,2} = x_{12,2} = x_{13,2} = x_{14,2} = 0$$

Here it is obvious that $\sum_{i=1}^{K_2} x_{i,2} \neq Q_2$ since $\sum_{i=1}^{K_2} x_{i,2} = 0$ and $0 \neq 17$. Thus, we need to use the fraction factor $f_{s,2}$ to recalculate the new idle time Z_2 as

$$Z_2 = Z_2 + P_{2,1} \min(1 - f_{s,2}) = 1.5x10^{-10} + (1 - 1.5x10^{-10}) = 1.$$

Now, when we recalculate the new sublot sizes using the new idle time $Z_2 = 1$, we obtain:

$$x_{1,2} = \min\left\{\left\lfloor [1-(1-7).0]/1 \right\rfloor, 17-0\right\} = \min\{1,17\} = 1$$

$$x_{2,2} = \min\left\{\left\lfloor [1-(1-7).1]/1 \right\rfloor, 17-1\right\} = \min\{7,16\} = 7$$

$$x_{3,2} = \min\left\{\left\lfloor [1-(1-7).8]/1 \right\rfloor, 17-8\right\} = \min\{79,9\} = 9$$

$$x_{4,2} = \min\left\{\left\lfloor [1-(1-7).17]/1 \right\rfloor, 17-17\right\} = 0$$

$$x_{4,2} = x_{5,2} = x_{6,2} = x_{7,2} = x_{8,2} = x_{9,2} = x_{10,2} = x_{11,2} = x_{12,2} = x_{13,2} = x_{14,2} = 0$$

Here, again we need to check whether that sum of the sublot sizes are equal to the total lot size. We observe that

$$\sum_{i=1}^{K_2} x_{i,2} = 1 + 7 + 9 = 17 = Q_2 \ .$$

When we analyze the sublot sizes, we see that 3 sublots are enough for Job 2. So the remaining 11 sublots will be transferred to Job 1, which is the job with the maximum process time after Job 2. Thus, the total number of sublots for Job 1 becomes 11+1=12. Now, we need to find integer sized sublots for Job 1. First, we need to calculate our sublot factor α_1 which is

$$\alpha_1 = \frac{P_{1,2}}{P_{2,1}} = \frac{6}{3} = 2.$$

When we calculate the sizes of each sublot for Job 1 using α_1, we obtain:

$$x_{1,1} = \left(\frac{2-1}{2^{12}-1}\right).12 = 12.93x10^{-3}$$

$$x_{2,1} = \alpha.x_{1,1} = 5.86x10^{-3}$$

$$x_{3,1} = \alpha^2.x_{1,1} = 0.0117$$

$$x_{4,1} = \alpha^3.x_{1,1} = 0.023$$

$$x_{5,1} = \alpha^4.x_{1,1} = 0.046$$

$$x_{6,1} = \alpha^5.x_{1,1} = 0.093$$

$$x_{7,1} = \alpha^6.x_{1,1} = 0.187$$

$$x_{8,1} = \alpha^7.x_{1,1} = 0.375$$

$$x_{9,1} = \alpha^8.x_{1,1} = 0.75$$

$$x_{10,1} = \alpha^9.x_{1,1} = 1.5$$

$$x_{11,1} = \alpha^{10}.x_{1,1} = 3$$

$$x_{12,1} = \alpha^{11}.x_{1,1} = 6$$

But it is obvious that sublot sizes are not integer sized and we need integer sized sublots. To do this, we need to calculate the idle time for Job 1 which is

$$Z_1 = \max_{1\le u\le K_1}\left\{\sum_{s=1}^{u} P_{1,1}.x_{s,1} - \sum_{s=1}^{u-1} P_{2,1}.x_{s,1}\right\} = 0.0087$$

Now by using this idle time Z_1 we calculate the non-integer sized sublots of Job 1 using the following formula

$$x_{s,n} = \min\left\{\left\lfloor \left[Z_n - (P_{n,1} - P_{n,2})\sum_{m=1}^{s-1} x_{m,n}\right] \Big/ P_{n,1} \right\rfloor, Q_n - \sum_{m=1}^{s-1} x_{m,n}\right\}.$$

Now, when we recalculate the integer sized sublots, we see all of them are zero. That is,

$$x_{1,1} = \min\left\{\left\lfloor [0.0087 - (3-6).0] \big/ 3 \right\rfloor, 12 - 0\right\} = \min\left\{\left\lfloor 2.9x10^{-3} \right\rfloor, 12\right\} = 0$$

$$x_{2,1} = \min\left\{\left\lfloor [0.0087 - (-3).0] \big/ 3 \right\rfloor, 12 - 0\right\} = \min\left\{\left\lfloor 0.0087 \right\rfloor, 12\right\} = 0$$

$$x_{3,1} = x_{4,1} = x_{5,1} = x_{6,1} = x_{7,1} = x_{8,1} = x_{9,1} = x_{10,1} = x_{11,1} = x_{12,1} = 0$$

Here it is obvious that $\sum_{i=1}^{K_1} x_{i,1} \neq Q_1$ since $\sum_{i=1}^{K_1} x_{i,1} = 0$ and $0 \neq 12$. Thus, we need to use the fraction factor $f_{s,1}$ to recalculate the new idle time Z_1 as

$$Z_1 = Z_1 + P_{1,1} \min(1 - f_{1,2}) = 2.99 + 0.0087 \cong 3.$$

Now, when we recalculate the new sublot sizes using the new idle time $Z_1 = 3$, we obtain:

$$x_{1,1} = \min\left\{\left\lfloor 3 - (3-6).0 \big/ 3 \right\rfloor, 12 - 0\right\} = \min\{2, 12\} = 2$$

$$x_{2,1} = \min\left\{\left\lfloor [3 - (3-6).2] \big/ 3 \right\rfloor, 12 - 2\right\} = \min\{3, 10\} = 3$$

$$x_{3,1} = \min\left\{\left\lfloor [1 - (-3).5] \big/ 3 \right\rfloor, 12 - 5\right\} = 6$$

$$x_{4,1} = \min\left\{\left\lfloor [3 - (-3).11] \big/ 3 \right\rfloor, 12 - 11\right\} = 1$$

$$x_{5,1} = \min\left\{\left\lfloor [3 - (-3).15] \big/ 3 \right\rfloor, 12 - 12\right\} = 0$$

$$x_{5,1} = x_{6,1} = x_{7,1} = x_{8,1} = x_{9,1} = x_{10,1} = x_{11,1} = x_{12,1} = 0$$

Here, again we need to check whether that sum of the sublot sizes are equal to the total lot size. We observe that

$$\sum_{i=1}^{K_1} x_{i,1} = 2 + 3 + 6 + 1 = 12 = Q_1.$$

When we analyze the sublot sizes, we see that 4 sublots are enough for Job 1. So the remaining 8 sublots will be transferred to Job 3, which is the job with the maximum process time after Job 1. Thus, the total number of sublots for Job 3 becomes 8+1=9. Now, we need to find integer sized sublots for Job 1. First, we need to calculate our sublot factor α_3 which is

$$\alpha_3 = \frac{P_{3,2}}{P_{3,1}} = \frac{4}{1} = 4.$$

When we calculate the sizes of each sublot for Job 3 using α_3, we obtain:

$$x_{1,3} = \left(\frac{4-1}{4^9-1}\right).16 = 1.83x10^{-4}$$

$$x_{2,3} = \alpha.x_{1,3} = 7.32x10^{-4}$$

$$x_{3,3} = \alpha^2.x_{1,3} = 02.92x10^{-3}$$

$$x_{4,3} = \alpha^3.x_{1,3} = 0.011$$

$$x_{5,3} = \alpha^4.x_{1,3} = 0.046$$

$$x_{6,3} = \alpha^5.x_{1,3} = 0.187$$

$$x_{7,3} = \alpha^6.x_{1,3} = 0.75$$

$$x_{8,3} = \alpha^7.x_{1,3} = 3$$

$$x_{9,3} = \alpha^8.x_{1,3} = 12$$

But it is obvious that sublot sizes are not integer sized and we need integer sized sublots. To do this, we need to calculate idle time for Job 3 which is

$$Z_3 = \max_{1 \le u \le K_3}\left\{\sum_{s=1}^{u} P_{3,1}.x_{s,3} - \sum_{s=1}^{u-1} P_{3,2}.x_{s,3}\right\} = 1.83x10^{-4}$$

Now by using this idle time Z_3 we calculate the non-integer sized sublots of Job 3 using the following formula.

$$x_{s,n} = \min\left\{\left\lfloor \left[Z_n - (P_{n,1} - P_{n,2})\sum_{m=1}^{s-1} x_{m,n}\right] \Big/ P_{n,1} \right\rfloor, Q_n - \sum_{m=1}^{s-1} x_{m,n}\right\}$$

Now, when we recalculate integer sized sublots, we see all of them are zero. That is,

$$x_{1,3} = \min\left\{\left\lfloor \left[1.83x10^{-4} - (-3).0\right] \Big/ 1 \right\rfloor, 16-0\right\} = 0$$

$$x_{2,3} = x_{3,3} = x_{4,3} = x_{5,3} = x_{6,3} = x_{7,3} = x_{8,3} = x_{9,3} = 0$$

Here it is obvious that $\sum_{i=1}^{K_3} x_{i,3} \neq Q_3$ since $\sum_{i=1}^{K_3} x_{i,3} = 0$ and $0 \neq 16$. Thus, we need to use the fraction factor $f_{s,3}$ to recalculate the new idle time Z_3 as

$$Z_3 = Z_3 + P_{3,1} \min(1 - f_{1,3}) = 1.83x10^{-4} + 1 - 1.83x10^{-4} = 1.$$

Now, when we recalculate new sublot sizes using new idle time $Z_3 = 1$, we obtain:

$$x_{1,3} = \min\left\{ \left\lfloor {1-(-3).0}/{1} \right\rfloor, 16-0 \right\} = 1 \qquad x_{3,3} = \min\left\{ \left\lfloor {[1-(-3).5]}/{1} \right\rfloor, 16-5 \right\} = 11$$

$$x_{2,3} = \min\left\{ \left\lfloor {[1-(-3).1]}/{1} \right\rfloor, 16-1 \right\} = 4 \qquad x_{4,3} = \min\left\{ \left\lfloor {[1-(-3).16]}/{1} \right\rfloor, 16-16 \right\} = 0$$

$$x_{4,3} = x_{5,3} = x_{6,3} = x_{7,3} = x_{8,3} = x_{9,3} = 0$$

Here, again we need to check whether that sum of the sublot sizes are equal to the total lot size. We observe that

$$\sum_{i=1}^{K_3} x_{i,3} = 1 + 4 + 11 = 16 = Q_3 .$$

When we analyze the sublot sizes, we see that 3 sublots are enough for Job 3. So the remaining 6 sublots will be transferred to Job 5, which is the job with the maximum process time after Job 3. Thus, the total number of sublots for Job 5 becomes 6+1=7. Now, we need to find integer sized sublots for Job 5. First, we need to calculate our sublot factor α_5 which is

$$\alpha_5 = \frac{P_{5,2}}{P_{5,1}} = \frac{6}{4} = 1.5 .$$

When we calculate the sizes of each sublot for Job 5 using α_5, we obtain:

$$x_{1,5} = \left(\frac{1.5-1}{1.5^7-1}\right).5 = 0.155$$

$$x_{2,5} = \alpha . x_{1,5} = 0.233$$

$$x_{3,5} = \alpha^2 . x_{1,5} = 0.349$$

$$x_{4,5} = \alpha^3 . x_{1,5} = 0.52$$

$$x_{5,5} = \alpha^4 . x_{1,5} = 0.78$$

$$x_{6,5} = \alpha^5 . x_{1,5} = 1.18$$

$$x_{7,5} = \alpha^6 . x_{1,5} = 1.77$$

But it is obvious that sublot sizes are not integer sized and we need integer sized sublots. To do this, we need to calculate the idle time for Job 5 which is

$$Z_5 = \max_{1 \le u \le K_5} \left\{ \sum_{s=1}^{u} P_{5,1} . x_{s,5} - \sum_{s=1}^{u-1} P_{5,2} . x_{s,5} \right\} = 0.621$$

Now by using this idle time Z_5 we calculate the non-integer sized sublots of Job 5 using the following formula

$$x_{s,n} = \min \left\{ \left\lfloor \left[Z_n - (P_{n,1} - P_{n,2}) \sum_{m=1}^{s-1} x_{m,n} \right] \Big/ P_{n,1} \right\rfloor, Q_n - \sum_{m=1}^{s-1} x_{m,n} \right\}.$$

Now, when we recalculate the integer sized sublots, we see all of them are zero. That is,

$$x_{1,5} = \min \left\{ \left\lfloor [0.621 - (-2).0] \Big/ 4 \right\rfloor, 5 - 0 \right\} = 0$$

$$x_{2,5} = x_{3,5} = x_{4,5} = x_{5,5} = x_{6,5} = x_{7,5} = 0$$

Here it is obvious that $\sum_{i=1}^{K_5} x_{i,5} \neq Q_5$ since $\sum_{i=1}^{K_5} x_{i,5} = 0$ and $0 \neq 5$. Thus, we need to use the fraction factor $f_{s,5}$ to recalculate the new idle time Z_5 as

$$Z_5 = Z_5 + P_{5,1} \min\left(1 - f_{1,5}\right) = 0.621 + \left(4x(1 - 0.1554)\right) = 0.621 + 3.378 = 3.9994 \cong 4 .$$

Now, when we recalculate the new sublot sizes using the new idle time $Z_5 = 4$, we obtain:

$$x_{1,5} = \min\left\{\left\lfloor {4 - (-2).0}\big/{4} \right\rfloor, 5 - 0\right\} = 1$$

$$x_{2,5} = \min\left\{\left\lfloor {[4 - (-2).1]}\big/{4} \right\rfloor, 5 - 1\right\} = 1$$

$$x_{3,5} = \min\left\{\left\lfloor {[4 - (-2).2]}\big/{4} \right\rfloor, 5 - 2\right\} = 2$$

$$x_{4,5} = \min\left\{\left\lfloor {[4 - (-2).5]}\big/{4} \right\rfloor, 5 - 5\right\} = 0$$

Here, when we calculate the rest of sublots sizes, we will see that

$$x_{5,5} = x_{6,5} = x_{7,5} = 0 .$$

Here, again we need to check whether that sum of the sublot sizes are equal to the total lot size. We observe that

$$\sum_{i=1}^{K_5} x_{i,5} = 1 + 1 + 2 + 1 = 5 = Q_5 .$$

When we analyze the sublot sizes, we see that 4 sublots are enough for Job 5. Thus, the remaining 3 sublots are useless, and 17 sublots are enough as the total number of sublots for all jobs.

Below, in Table 4-3, the sizes and process times of each integer sized sublot is given.

Table 4-3 Size and Process Times of Integer Sized Sublots

SUBLOT NO	Sublot Size	Total Sublot Process Time on M1	Total Sublot Process Time on M2
$x_{1,1}$	2	3	6
$x_{1,2}$	3	3	6
$x_{1,3}$	6	3	6
$x_{1,4}$	1	3	6
$x_{2,1}$	1	1	7
$x_{2,2}$	7	1	7
$x_{2,3}$	9	1	7
$x_{3,1}$	1	1	4
$x_{3,2}$	4	1	4
$x_{3,3}$	11	1	4
$x_{4,1}$	1	1	9
$x_{4,2}$	9	1	9
$x_{4,3}$	10	1	9
$x_{5,1}$	1	4	6
$x_{5,2}$	1	4	6
$x_{5,3}$	2	4	6
$x_{5,4}$	1	4	6

As shown in Table 4-3, each sublot's process time on M1 is less than its process time on M2. So all sublots belongs to Set 1, and Set 2 is empty, obviously. That is,

$$Set\ 1 = \left\{ \begin{matrix} x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}, x_{2,1}, x_{2,2}, x_{2,3}, x_{3,1}, \\ x_{3,2}, x_{3,3}, x_{4,1}, x_{4,2}, x_{4,3}, x_{5,1}, x_{5,2}, x_{5,3}, x_{5,4} \end{matrix} \right\}.$$

Now, to find the sequence of sublots that provides the optimal makespan, we determine $OSet\ 1$ and $OSet\ 2$. While $Set\ 2$ is empty, $OSet\ 2$ is empty, too. But to obtain $OSet\ 1$, we rearrange $Set\ 1$ as

$$OSet\ 1 = \begin{Bmatrix} x_{2,1}, x_{3,1}, x_{4,1}, x_{1,1}, x_{3,2}, x_{5,1}, x_{5,2}, x_{5,4}, \\ x_{1,2}, x_{2,2}, x_{5,3}, x_{2,3}, x_{4,2}, x_{4,3}, x_{3,3}, x_{1,3}, x_{1,4} \end{Bmatrix}.$$

Thus, we obtain the sizes and sequence of unequal sublots that provides smallest makespan using our heuristic algorithm for the lot streaming problem under consideration.

4.2 Heuristic Algorithm for the Case with Equal Sized Sublots

With this heuristic, we aim to reach the minimum makespan for the multi-product lot streaming problem with equal sized sublots. To address this problem, we develop two-parted heuristic that each part contains first heuristics basically. At the first part of the heuristic, we consider each lots of job. Sequentially, we assign the max number of sublots to each job and find the number of the sublots for each product, the size of each sublot on each machine and the sequence for processing the sublots. We pick the one with minimum makespan as a result. At the second part of the heuristic; we work on the two jobs with the highest processing times and again we find the number of the sublots for each product, the size of each sublot on each machine and the sequence for processing the sublots according to our heuristic. At the end, heuristic returns the solution of part that has the minimum makespan time.

Below, we present the notation and formulation of the two-stepped heuristic algorithm:

Parameters:

M	Number of machines ($M=2$)
N	Number of jobs
S	Total number of sublots
Q_n	Lot size of job n
$P_{n,m}$	Processing time for one unit of job n on machine m
$TP_{n,m}$	Total processing time for one unit of job n on machine m
K_n	Number of sublots of job n
$x_{s,n}$	Size of the sublot s of job n
OS	Optimal sublot sequence
LOS	List of optimal sublot sequence
$TSP_{n,,s,m}$	Total processing time for sublot s of job n on machine m

Indices:

m	Machine index where $m=1,...,M$
n	Job index where $n=1,...,N$
s	Sublot index where $s=1,...,K_n$

Heuristic Algorithm for Equal Sized Sublots Case:

Step 1: (Sequentially, Split All Lots of Jobs Equally)

Step 1.1:

For each job J_n,

Step 1.1.1:

Identify number of sublot used by each job. We assume that J_n is the job with the highest $TP_{n,m}$ value. We assume that each job except J_n value has only one sublot and the rest of the sublots belong to J_n.

$$K_n = \begin{cases} 1 \quad if \;\; job\, n\, is\, not\, J_n \\ S - (N - 1)\, otherwise \end{cases}$$

Step 1.1.2:

Find the size of each equal-sized sublot of J_n as

$$x_{s,n} = \frac{Q_n}{K_n}.$$

The jobs except J_n aren't split. For these jobs sublot size equals to lot size, i.e.

$$x_{1,n} = Q_n \text{ where } n \text{ is NOT the } J_n.$$

Step 1.1.3:

Calculate the total process time of each sublot ($TSP_{n,,s,m}$) on both machine and group them as *Set 1* and *Set 2*.

Set 1 and *Set 2* are mathematically expressed as

$$Set\ 1 = \{ k_n : TSP_{n,s,1} \le TSP_{n,s,2} \}$$

$$Set\ 2 = \{ k_n : TSP_{n,s,1} \ge TSP_{n,s,2} \}$$

where $TSP_{n,,s,m} = x_{s,n}.P_{n,m}$, $n \in [1, N]$, $k_n \in [1, K_n]$ and $m = \{1,2\}$.

Step 1.1.4:

Optimize *Set 1* and *Set 2* by rearranging their entities. While *Set 1* is optimized by sorting sublots according to their increasing process time on M_1; *Set 2* is optimized by sorting sublots according to decreasing process time on M_2. If we call optimized *Set 1* as *OSet 1* and optimized *Set 2* as *OSet 2,* the mathematical representation of *OSet 1* and *OSet 2* are as follows:

$OSet\ 1 = \{\forall k_n \in Set1 : TSP_{n,s,1} \leq TSP_{n+1,s,1}\}$

$OSet\ 2 = \{\forall k_n \in Set2 : TSP_{n,s,2} \geq TSP_{n+1,s,2}\}$ where $n \in [1, N], k_n \in [1, K_n]$

Step 1.1.5:

To calculate makespan for J_n, optimized sublot sequence is needed. Optimized sublot sequence for J_n, OS_n, is a combination of *Set 1* and *Set 2* as all elements of *Set 1* is followed by all elements of *Set 2*. OS_n is represented mathematically as follows:

$$OS_n = \bigcup_{r=1}^{S_1} k_r \cup \bigcup_{m=1}^{S_2} k_m$$

where $k_r \in Set1$, $k_m \in Set2$, S_1 and S_2 are the sizes of *Set 1* and *Set 2* of J_n, respectively.

After calculating OS_n, we add OS_n to LOS.

Step 1.2:

After splitting all jobs into equal sublots and adding optimal sublot sequence of each job to LOS, we find the sequence on LOS with the smallest makespan. This sequence is the optimal sublot sequence obtained in Step 1. That is,

$$OS = \min \{OS_n; 1 \leq n \leq N\}.$$

Step 2: (Split Two Jobs with the First and Second Highest Processing Times)

Step 2.1:

Identify the maximum process time on M_1 and M_2, that is:

$$\exists n, m\,,\ max\left\{TP_{n,m}\right\} \text{ where } TP_{n,m} = P_{n,m}.Q_n$$

The two of jobs with the highest $TP_{n,m}$ value will be the primary and the secondary jobs to calculate sublot sizes. Let's call the first job (primary job) with highest $TP_{n,m}$ value JH_1; and the second job (secondary job) with highest $TP_{n,m}$ value JH_2.

Step 2.2:

Identify number of sublot used by each job. We assume that the jobs except JH_1 and JH_2 have only one sublot; JH_2 has two sublots and JH_1 has the rest of the sublots.

$$\mathrm{K_n} = \begin{cases} 1 \;\; \textit{if} \text{ job } n \text{ is neither the index of } JH_1 \text{ nor } JH_2 \\ 2 \;\; \textit{if} \text{ job } n \text{ is the index of } JH_2 \\ \mathrm{S - (N - 1) - 1} \;\; \textit{if} \text{ job } n \text{ is the index of } JH_1 \end{cases}$$

Step 2.3:

Find the size of each equal-sized sublot of JH_1 and JH_2 as

$$x_{s,n} = \frac{Q_n}{K_n}.$$

Obviously, the jobs except JH_1 and JH_2 the sublot size of job equals to lot size of job; because they are not split.

$$x_{1,n} = Q_n \text{ where } n \text{ is NOT the } J_n.$$

Step 2.4:

Calculate the total process time of each sublot ($TSP_{n,,s,m}$) on both machine and group them as *Set 1* and *Set 2*.

Set 1 is a set of sublots that are processed on M_1 at most in the time that are processed on M_2. In other words; each sublot of *Set 1* is processed on M_1 in less or equal time on M_2.

Set 2 is a set of sublots that are processed on M_1 longer than that are processed on M_2.

Set 1 and *Set 2* are mathematically expressed as

$$Set\ 1 = \{ k_n : TSP_{n,s,1} \leq TSP_{n,s,2} \}$$

$$Set\ 2 = \{ k_n : TSP_{n,s,1} \geq TSP_{n,s,2} \}$$

where $TSP_{n,,s,m} = x_{s,n} P_{n,m}$, $n \in [1,N]$, $k_n \in [1,K_n]$ and $m = \{1,2\}$.

Step 2.5:

Optimize *Set 1* and *Set 2* by rearranging their entities. While *Set 1* is optimized by sorting sublots according to their increasing process time on M_1; *Set 2* is optimized by sorting sublots according to decreasing process time on M_2. If we call optimized *Set 1* as *OSet 1* and optimized *Set 2* as *OSet 2,* the mathematical representation of *OSet 1* and *OSet 2* are as follows:

$$OSet\ 1 = \{ \forall k_n \in Set1 : TSP_{n,s,1} \leq TSP_{n+1,s,1} \}$$

$$OSet\ 2 = \{ \forall k_n \in Set2 : TSP_{n,s,2} \geq TSP_{n+1,s,2} \} \text{where } n \in [1,N], k_n \in [1,K_n]$$

Step 2.6:

To minimize makespan, sublot sequence is needed to be optimized. Optimized sublot sequence, *OS,* is a combination of *Set1* and *Set2* as all elements of *Set1* is followed by all elements of *Set2*. *OS* is represented mathematically as follows:

$$OS = \bigcup_{n=1}^{S_1} k_n \cup \bigcup_{m=1}^{S_2} k_m$$

where, $k_n \in Set\,1$, $k_m \in Set\,2$, S_1 and S_2 are the sizes of *Set 1* and *Set 2* of J_n, respectively.

Step 3:

If the makespan of the sequence obtained in Step 1 is smaller than that of the one obtained in Step 1, then the solution obtained in Step 1 should be selected for implementation. Otherwise; the solution obtained in Step 2 should be selected.

4.2.1 Numerical Example

In this section, we provide a numerical example for illustrating the heuristic algorithm for solving the equal sized sublot case. Consider the simple instance of the problem given in Section 4.1.1 in which there are 5 jobs and the total number of sublots is 20. Unit processing times on the machines and the lot sizes for all jobs are given in Table 4-4.

Table 4-4 Process Times on M1 and M2

JOB NO	Process Time on M1	Process Time on M2	Total Lot Size
Job 1	3	6	12
Job 2	1	7	17
Job 3	1	4	16
Job 4	1	9	20
Job 5	4	6	5

First we will find the job with the maximum process time. As seen on Table 4-1, the job with the maximum process time is Job 4. So, Job 4 will have the maximum number of sublots when other jobs have only one sublot. It means that that Job 4 will be split into 16 equal sized sublots while the other jobs will not split into sublots. That is,

$$K_4 = 20 - 4 = 16 \text{ and } K_1 = K_2 = K_3 = K_5 = 1.$$

Now, we shall find the sizes of sublots for each job. Here it is obvious that for Jobs 1, 2, 3 and 5 the sublot size equals to the total lot size of the job since these jobs are not split into sublots and have only one sublot, which is equal to the lot. But Job 4 is split in to 16 sublots. Size of each sublot in Job 4 is

$$x_{s,4} = \frac{Q_4}{K_4} = \frac{20}{16} = 1.25 \text{ where } 1 \le s \le 16 .$$

So by splitting only Job 4's total lot, we obtain the sublots sizes in Table 4-5.

Table 4-5 Sublot Sizes and Process Times Obtained in Step 1

SUBLOT NO	Sublot Size	Total Sublot Process Time on M1	Total Sublot Process Time on M2
$x_{1,1}$	12	36	72
$x_{1,2}$	17	17	119
$x_{1,3}$	16	16	64
$x_{1,4}$	1.25	1.25	11.25
$x_{2,4}$	1.25	1.25	11.25
$x_{3,4}$	1.25	1.25	11.25
$x_{4,4}$	1.25	1.25	11.25
$x_{5,4}$	1.25	1.25	11.25
$x_{6,4}$	1.25	1.25	11.25
$x_{7,4}$	1.25	1.25	11.25
$x_{8,4}$	1.25	1.25	11.25
$x_{9,4}$	1.25	1.25	11.25
$x_{10,4}$	1.25	1.25	11.25
$x_{11,4}$	1.25	1.25	11.25
$x_{12,4}$	1.25	1.25	11.25
$x_{13,4}$	1.25	1.25	11.25
$x_{14,4}$	1.25	1.25	11.25
$x_{15,4}$	1.25	1.25	11.25
$x_{16,4}$	1.25	1.25	11.25
$x_{1,5}$	5	20	30

As seen on Table 4-5, all sublots belong to *Set 1* since the total process time on M1 is less than the total process time on M2 for all jobs. That is,

$$Set\,1 = \left\{ \begin{array}{l} x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}, x_{2,4}, x_{3,4}, x_{4,4}, x_{5,4}, x_{6,4}, x_{7,4}, \\ x_{8,4}, x_{9,4}, x_{10,4}, x_{11,4}, x_{12,4}, x_{13,4}, x_{14,4}, x_{15,4}, x_{16,4}, x_{1,5} \end{array} \right\}$$

To find the sequence of sublots that provides the optimal makespan, we determine $OSet\ 1$ by rearranging it as

$$OSet\ 1 = \left\{ \begin{array}{l} x_{1,4}, x_{2,4}, x_{3,4}, x_{4,4}, x_{5,4}, x_{6,4}, x_{7,4}, \\ x_{8,4}, x_{9,4}, x_{10,4}, x_{11,4}, x_{12,4}, x_{13,4}, x_{14,4}, x_{15,4}, x_{16,4}, x_{1,3}, x_{1,2}, x_{1,5}, x_{1,1} \end{array} \right\}$$

Thus, the optimal sequence of sublots obtained by splitting Job 4 into 16 sublots is equal to $OSet\ 1$. That is,

$$OS_4 = \begin{Bmatrix} x_{1,4}, x_{2,4}, x_{3,4}, x_{4,4}, x_{5,4}, x_{6,4}, x_{7,4}, \\ x_{8,4}, x_{9,4}, x_{10,4}, x_{11,4}, x_{12,4}, x_{13,4}, x_{14,4}, x_{15,4}, x_{16,4}, x_{1,3}, x_{1,2}, x_{1,5}, x_{1,1} \end{Bmatrix}$$

If we apply the same procedure for Jobs 2, 3, 1 and 5, we obtain the following optimal sequences of sublots obtained by splitting Jobs 2, 3, 1 and 5 into 16 sublots, respectively. That is,

$$OS_2 = \begin{Bmatrix} x_{1,2}, x_{2,2}, x_{3,2}, x_{4,2}, x_{5,2}, x_{6,2}, x_{7,2}, x_{8,2}, x_{9,2}, x_{10,2}, \\ x_{11,2}, x_{12,2}, x_{13,2}, x_{14,2}, x_{15,2}, x_{16,2}, x_{1,3}, x_{1,4}, x_{1,5}, x_{1,1} \end{Bmatrix}$$

$$OS_3 = \begin{Bmatrix} x_{1,3}, x_{2,3}, x_{3,3}, x_{4,3}, x_{5,3}, x_{6,3}, x_{7,3}, x_{8,3}, x_{9,2}, x_{10,2}, \\ x_{11,3}, x_{12,3}, x_{13,3}, x_{14,3}, x_{15,3}, x_{16,3}, x_{1,2}, x_{1,4}, x_{1,5}, x_{1,1} \end{Bmatrix}$$

$$OS_1 = \begin{Bmatrix} x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1}, x_{5,1}, x_{6,1}, x_{7,1}, x_{8,1}, x_{9,1}, x_{10,1}, \\ x_{11,1}, x_{12,1}, x_{13,1}, x_{14,1}, x_{15,1}, x_{16,1}, x_{1,3}, x_{1,2}, x_{1,4}, x_{1,5} \end{Bmatrix}$$

$$OS_5 = \begin{Bmatrix} x_{1,5}, x_{2,5}, x_{3,5}, x_{4,5}, x_{5,5}, x_{6,5}, x_{7,5}, x_{8,5}, x_{9,5}, x_{10,5}, \\ x_{11,5}, x_{12,5}, x_{13,5}, x_{14,5}, x_{15,5}, x_{16,5}, x_{1,3}, x_{1,2}, x_{1,4}, x_{1,1} \end{Bmatrix}$$

Makespan values of the sequences obtained above are as follows:

$$MOS_4 = 466.25$$

$$MOS_2 = 466.0625$$

$$MOS_3 = 498$$

$$MOS_1 = 467.25$$

$$MOS_5 = 466.25$$

It is obvious that MOS_2 has the smallest makespan value. Thus, at the first step of the heuristic algorithm we obtain the following sublot sequence with a makespan value of 466.025 time units.

$$OS_2 = \begin{Bmatrix} x_{1,2}, x_{2,2}, x_{3,2}, x_{4,2}, x_{5,2}, x_{6,2}, x_{7,2}, x_{8,2}, x_{9,2}, x_{10,2}, \\ x_{11,2}, x_{12,2}, x_{13,2}, x_{14,2}, x_{15,2}, x_{16,2}, x_{1,3}, x_{1,4}, x_{1,5}, x_{1,1} \end{Bmatrix}$$

Now we continue with second step of the heuristic algorithm in which two jobs with the first and second highest processing times are only split into sublots. As seen on Table 4-4, the job with the maximum process time is Job 4 and the job with the second maximum process time is Job 2. According our heuristic approach, jobs except Job 4 and Job 2 will have only one sublot, which is equal to the lot, Job 2 will have 2 sublots, and Job 4 has the rest of the sublots. That is,

$$K_1 = K_3 = K_5 = 1$$

$$K_2 = 2$$

$$K_4 = 20 - 5 = 15$$

Now, we shall find the sizes of sublots for each job. Here it is obvious that for Jobs 1, 3 and 5 the sublot size equals to the total lot size of the job since these jobs are not split into sublots and have only one sublot, which is equal to the lot. But Job 2 is split in to 2 equal sized sublots. Size of each sublot in Job 2 is

$$x_{s,2} = \frac{Q_2}{K_2} = \frac{17}{2} = 8.5 \text{ where } 1 \le s \le 2\,.$$

For Job 4, we shall use 15 equal sublots. Size of each sublot in Job 4 is

$$x_{s,4} = \frac{Q_4}{K_4} = \frac{20}{15} = 1.33 \text{ where } 1 \le s \le 15\,.$$

Sublot sizes and process times of all jobs are as shown in Table 4-6.

Table 4-6 Sublot Sizes and Process Times Obtained in Step 2

SUBLOT NO	Sublot Size	Total Sublot Process Time on M1	Total Sublot Process Time on M2
$x_{1,1}$	12	36	72
$x_{1,2}$	8.5	25.5	51
$x_{2,2}$	8.5	25.5	51
$x_{1,3}$	16	16	64
$x_{1,4}$	1.33	1.33	11.97
$x_{2,4}$	1.33	1.33	11.97
$x_{3,4}$	1.33	1.33	11.97
$x_{4,4}$	1.33	1.33	11.97
$x_{5,4}$	1.33	1.33	11.97
$x_{6,4}$	1.33	1.33	11.97
$x_{7,4}$	1.33	1.33	11.97
$x_{8,4}$	1.33	1.33	11.97
$x_{9,4}$	1.33	1.33	11.97
$x_{10,4}$	1.33	1.33	11.97
$x_{11,4}$	1.33	1.33	11.97
$x_{12,4}$	1.33	1.33	11.97
$x_{13,4}$	1.33	1.33	11.97
$x_{14,4}$	1.33	1.33	11.97
$x_{15,4}$	1.33	1.33	11.97
$x_{1,5}$	5	20	30

As seen on Table 4-6, all sublots belong to *Set 1*. So *Set 2* is empty. That is,

$$Set\ 1 = \left\{ \begin{array}{l} x_{1,1}, x_{1,2}, x_{2,2}, x_{1,3}, x_{1,4}, x_{2,4}, x_{3,4}, x_{4,4}, x_{5,4}, x_{6,4}, x_{7,4}, \\ x_{8,4}, x_{9,4}, x_{10,4}, x_{11,4}, x_{12,4}, x_{13,4}, x_{14,4}, x_{15,4}, x_{1,5} \end{array} \right\}$$

To find the sequence of sublots that provides the optimal makespan, we determine $OSet\ \ 1$ by rearranging it as

$$OSet\ 1 = \left\{ \begin{array}{l} x_{1,4}, x_{2,4}, x_{3,4}, x_{4,4}, x_{5,4}, x_{6,4}, x_{7,4}, x_{8,4}, x_{9,4}, x_{10,4}, \\ x_{11,4}, x_{12,4}, x_{13,4}, x_{14,4}, x_{15,4}, x_{1,3}, x_{1,5}, x_{1,2}, x_{2,2}, x_{1,1} \end{array} \right\}.$$

CHAPTER 5

A SOFTWARE PACKAGE FOR SOLVING THE PROBLEMS

When we create heuristic method, we also create software for user to create test scenarios and reloading them to compare the result with mathematical model's result. This software is able to create scenarios depending on your choice of total number of sublots, total number of jobs, sublot division approach etc.; runs heuristic approach according to your heuristic choice; gives results in tabular and documented forms and create a reloading file for user to reloading obtained heuristic algorithm's result to compare mathematical model's one.

In this chapter, we briefly explain the implementation of this software and usage of it.

5.1 Brief Details of the Software Implementation

This software is created for users to run and see the results of heuristic algorithms in more user friendly way. Because of this software request, we coded in C# using .NET technologies.

When we created user interfaces, we used DevExpress library which is a gui library that creates ASP.NET UI controls that coder can use this library in his application to enhance UI.

When we coding the user interface for heuristic algorithms, we remain loyal to object oriented programing basis. In Figure 5.1, you can see the package diagram of this software.

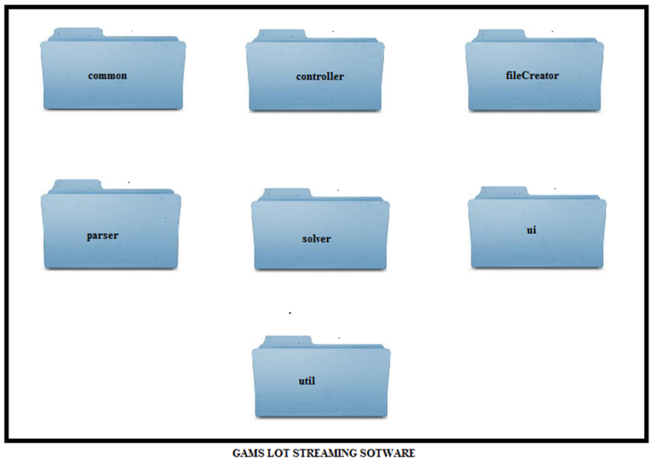


Figure 5.1 Package Diagram for Software

Brief content explanations of these packages are as follows:

common: This package contains classes that are used by commonly all packages. The class details of this package are given in Figure 5.2.

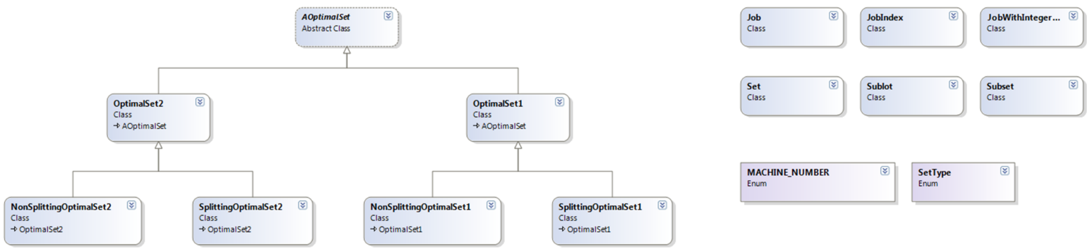


Figure 5.2 Class Diagram of Package "common"

controller: This package contains the classes to control sublot sizes according to the selected heuristic approach. The details of this package are illustrated in Figure 5.3.

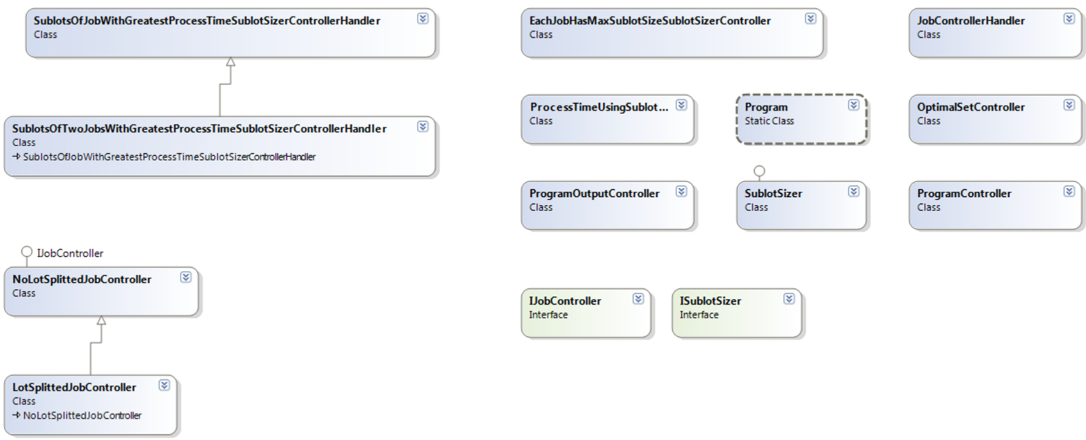


Figure 5.3 Class Diagram of Package "controller"

fileCreator: This package is responsible for creating "*.dat" files and "*.doc" files. "*.dat" files are used for reloading the heuristic results to the software to compare the mathematical model's results. "*.doc" files are detailed documentation of results. The class content of this package is shown in Figure 5.4.

Figure 5.4 Class Diagram of Package "fileCreator"

parser: This package contains the classes that are used for parsing "*.dat" files. By this way, user may reach previously run heuristic results again. The classes of this package are illustrated in Figure 5.5.

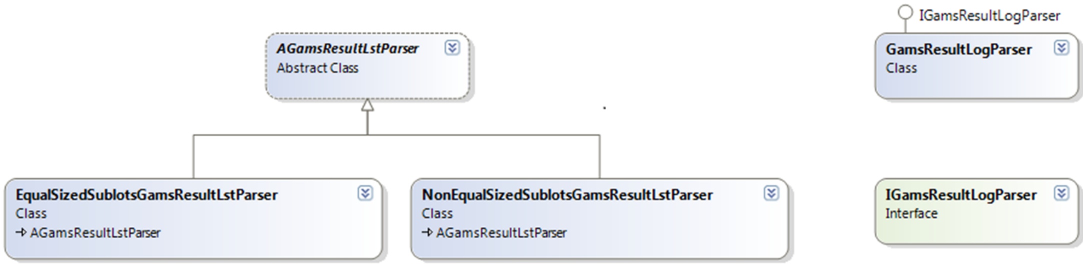


Figure 5.5 Class Diagram of Package "parser"

solver: This package is responsible for running the heuristic algorithms and generating results according to the user's choices. The class detail of this package is given in Figure 5.6.

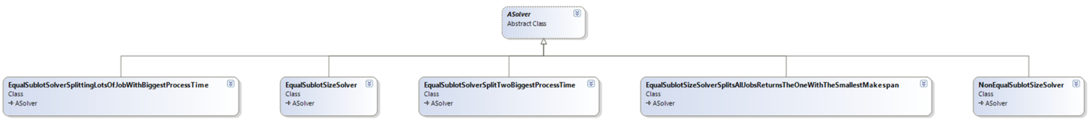


Figure 5.6 Class Diagram of Package "solver"

ui: This package contains all the windows that are used by the user himself. The classes of this package are shown in Figure 5.7.

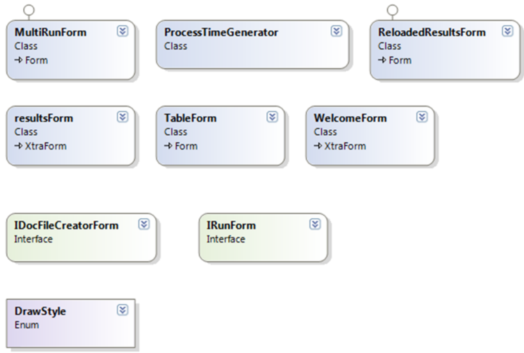


Figure 5.7 Class Diagram of Package "ui"

util: This package is utility package as it is understood by its name. It contains all utility functions. The class content of this package is shown in Figure 5.8.

Figure 5.8 Class Diagram of Package "util"

5.2 Software Usage

In this section, we briefly explain how the software is used. When the user runs the software, the welcome window, shown in Figure 5.9, appears. This window is for the user to make selection. If the user wants to create a scenario by entering the total number of jobs, total number of sublots etc., **Multi Run** choice item is selected. If the user wants to reload the multi run result, **Reload Result** item must be selected.

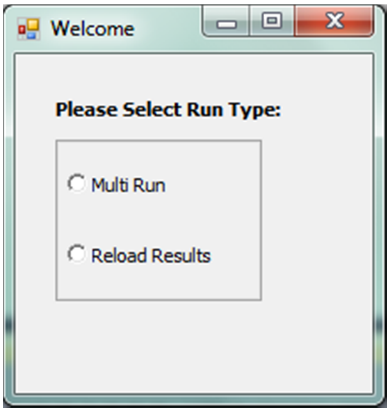


Figure 5.9 Welcome Window

If the user select **Multi Run** choice, the window in Figure 5.10, appears. This window is created for the user to create a scenario data and run heuristic algorithm. Using **Sublot Sizer** box, user chose how he want to split lots. If user wants to split lots equally, **Equal Sized Sublots** choice should be clicked. On the other hand, if the user wants to split lots unequally, **Non-Equal Sized Sublots** radio button has to be clicked.

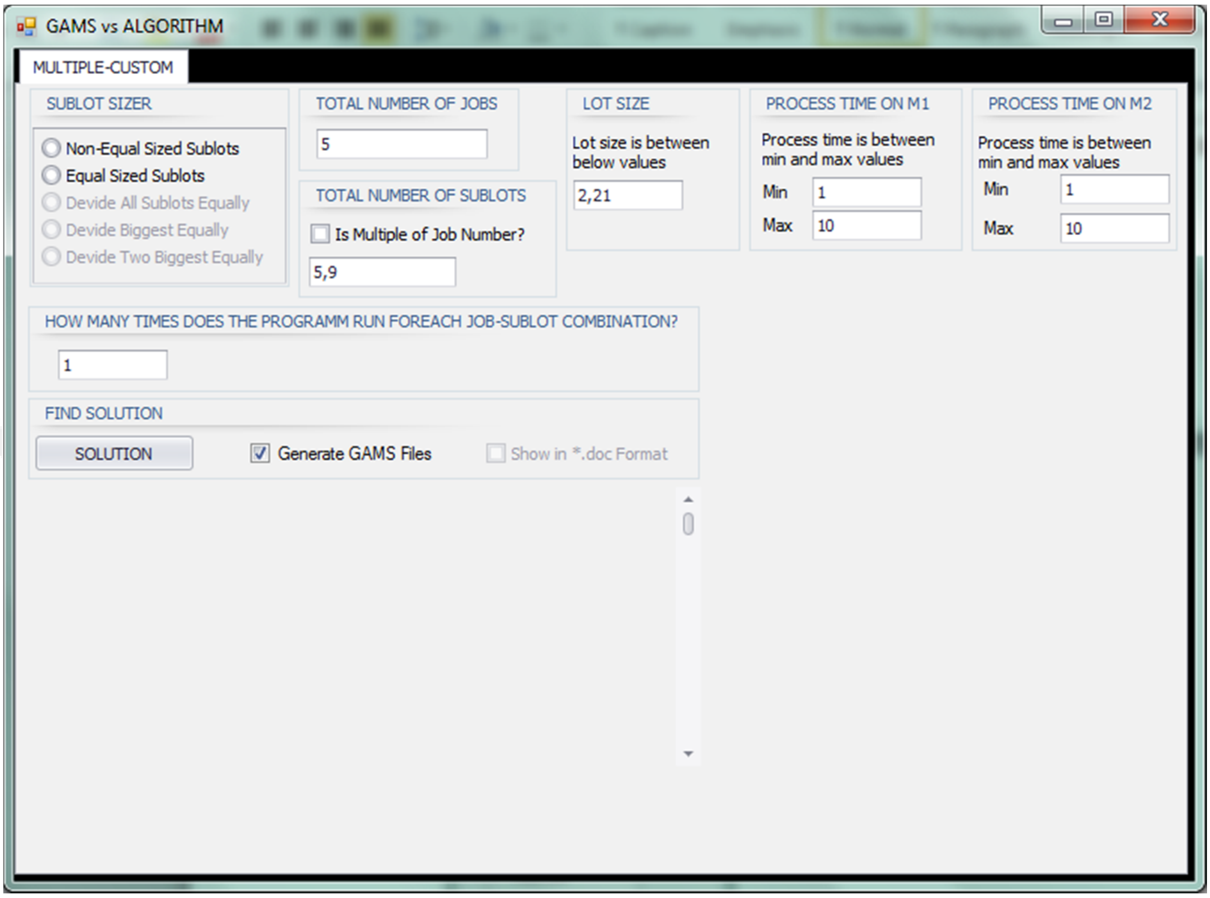


Figure 5.10 Multi Run Window

As you seen in Figure 5-10, multi run window is opened with default values. If the user wants to change the total number of jobs; **Total Number of Jobs** text box must be set. If the total number of sublots is wanted to be set, **Total Number of Sublots** text box is updated. If the user wants the total number of sublots as multiple of total number jobs, as we do at our scenarios, then it is enough to check **Is Multiple of Job Number** check box. Otherwise, the user may enter the total number of sublots by separating them with comma as 5, 10, 15 etc. In this case for each job-sublot pair, the heuristic is rerun and create scenario as combination of this entire pair. If user wants to set the total lot size, **Lot Size** text field must be used. User may enter maximum and minimum values of the total lot size by separating them with comma and code randomly generate the total lot size between these maximum and minimum values. But again if the user wants the lot sizes as a multiple of the total number of sublots, special notation may be used for this field. For example, if the user wants the total lot size as threefold of the total number of sublots, it is enough to set the lot size text field 3x. In the same way, if the user wants to test the heuristic algorithm for each job-sublot pair with the lot sizes that are twofold and threefold of the total number of sublots; it is enough to set this field 2x, 3x. For setting the process time for each unit of a job on machines 1 and 2, **Process Time on M1** and **Process Time on M2** values are set. In these fields, the user is allowed to set minimum and maximum values of process times and, code randomly generate process time for each unit of a job for each machine. Also, there exists a text box called **How Many Times Does the Program Run for Each job-Sublot Combination**. This value is used for generating different job-sublot pairs with newly generated lot sizes and process times. Also there is a checkbox called **Generate GAMS Files** that is used for automatically generating the GAMS code into user desktop to run the mathematical model for each job-sublot pair. After completing all these data entrance, **Solution** button is clicked to run the heuristic with these data.

As illustrated in Figure 5.11, for each job-sublot pair Gantt charts are created after the solution button is clicked. The details of Gantt chart are given in Figure 5.12. In Figure 5.12, each sublot is represented with different color and big rectangular boxes. The pop-up balloons on each sublot represent the sizes of corresponding sublots. Also when the square small boxes on the left side of each sublot represent the corresponding sublots' processing start up time; the right hand side one's represent the corresponding sublots' process completion time on that machine. At this step if the user want to see or save the data details, performance details, deviation comparisons of heuristic approach with mathematical model or just makespan values in "*.doc" format, **Show in *.doc** format checkbox is clicked.

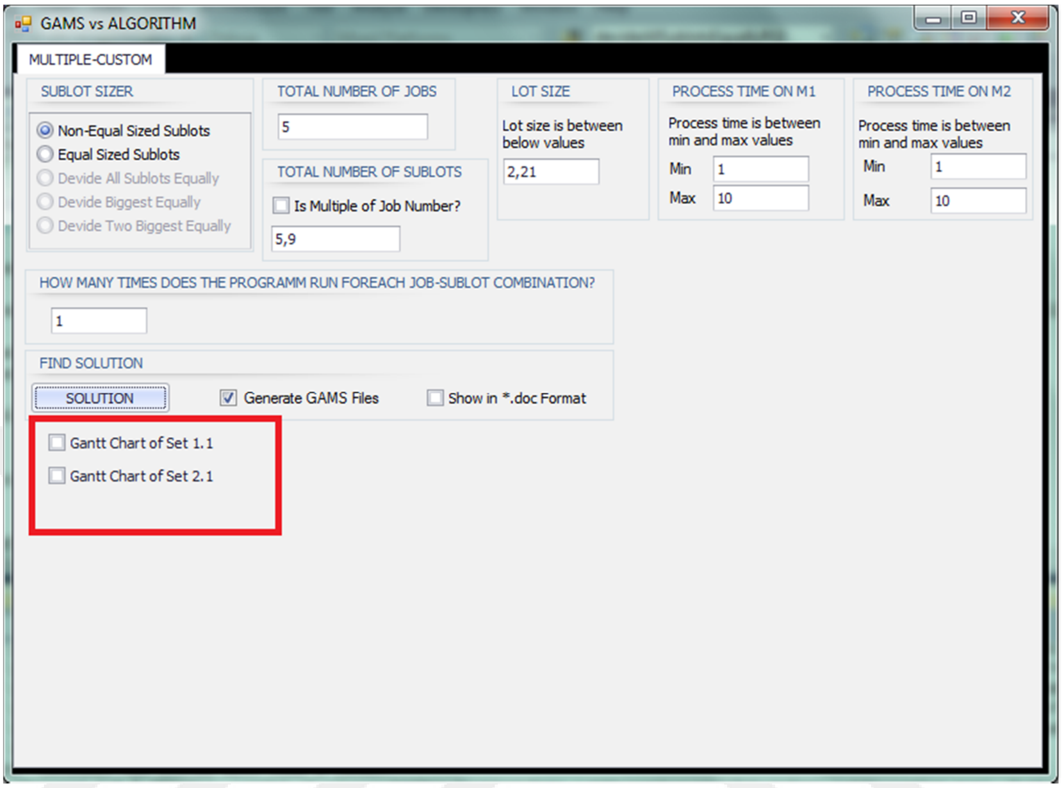


Figure 5.11 Multi Run Form after Solution is Generated

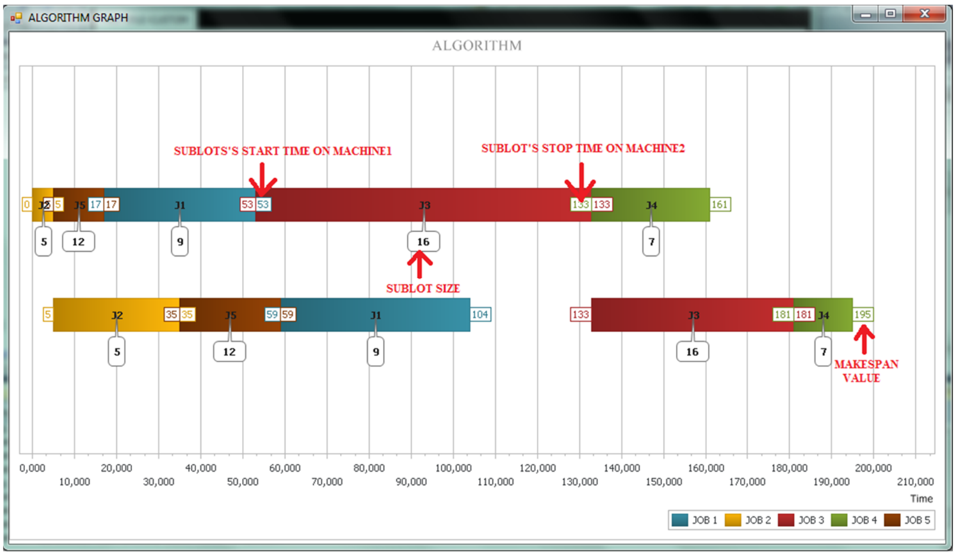


Figure 5.12 Gantt Chart Created by the Heuristic Algorithm

If the user wants to compare the results of the heuristic algorithm with the results of the mathematical model, **Reload Results** radio button in Figure 5.9 should be selected and **Reloaded Results Form** appears as shown in Figure 5.13.

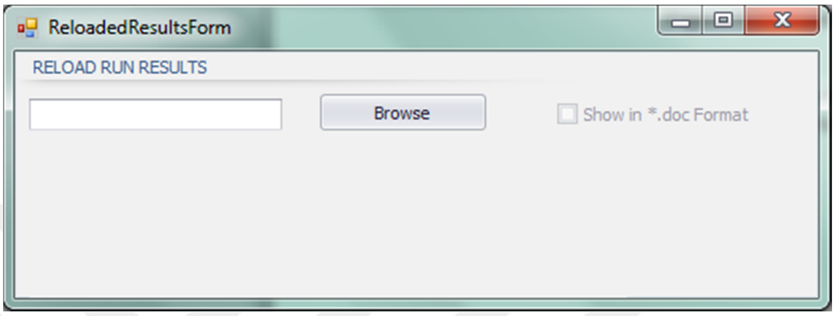


Figure 5.13 Reloaded Results Form

Reloaded Results Form is opened empty as seen on Figure 5.13 and waits the user to select the heuristic results that are generated by the heuristic itself, "*.dat" files. After the user clicks the **Browse** button and selects the dat files, the content of dat file is parsed and reloaded this form as shown in Figure 5.14.

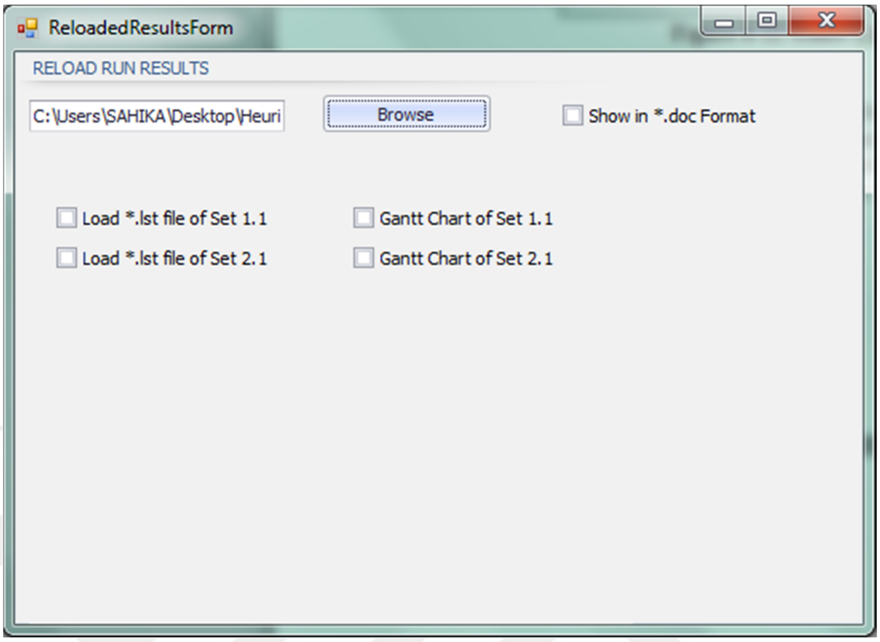


Figure 5.14 Reloaded Results Form after Dat Files are Reloaded

As seen in Figure 5.14, the user may load lst files. If the user load lst files, both Gantt charts obtained by the heuristic algorithm and the mathematical model solved by GAMS are shown and the user may see not only details of the heuristic results, but also the details of mathematical model results. For example, if we reload the results of the first set to this form, the lst file is parsed and the details are shown as in Figure 5.15.

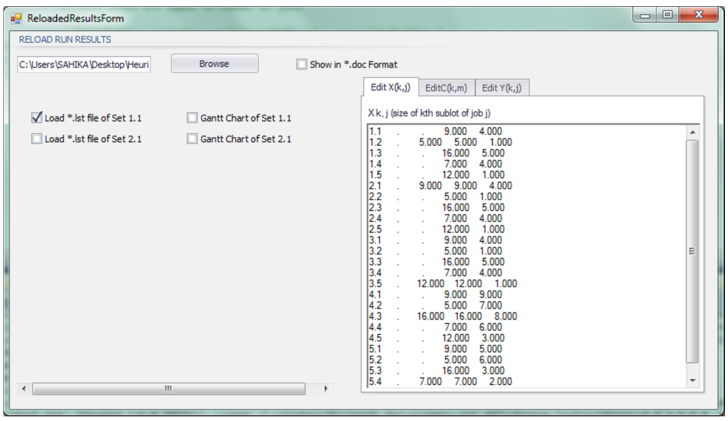


Figure 5.15 Reload Result Form after Lst File is Reloaded

Here, if the user clicks on Gantt Chart of Set1.1 checkbox, Gantt charts obtained by the heuristic algorithm and the mathematical model solved by GAMS are obtained as in Figure 5.16.Again when creating doc file at the Set1.1 section, the details of the results obtained by the heuristic algorithm and the mathematical model solved by GAMS are analyzed.

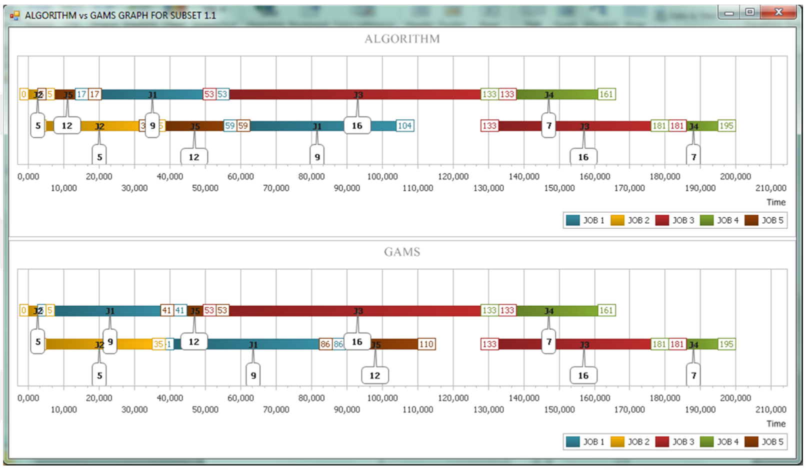


Figure 5.16 Gantt Charts Obtained by the Heuristic Algorithm and the Mathematical Model Solved by GAMS

CHAPTER 6

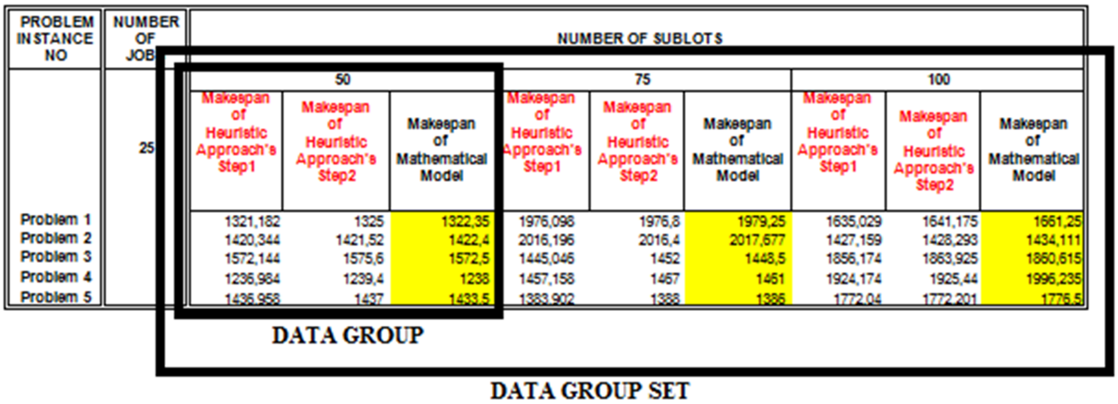
COMPUTATIONAL EXPERIMENTS

In this chapter, we describe our computational experiments to evaluate the effectiveness and efficiency of the MILP models and the proposed heuristic algorithms in solving the MPLS problem under consideration.

The mathematical models are coded and solved in GAMS 23.7. All computational experiments are conducted on laptop with Intel Core i5 with 2.30 GHz CPU and 4GB RAM under 64-bit Windows Home 7 Premium operating system.

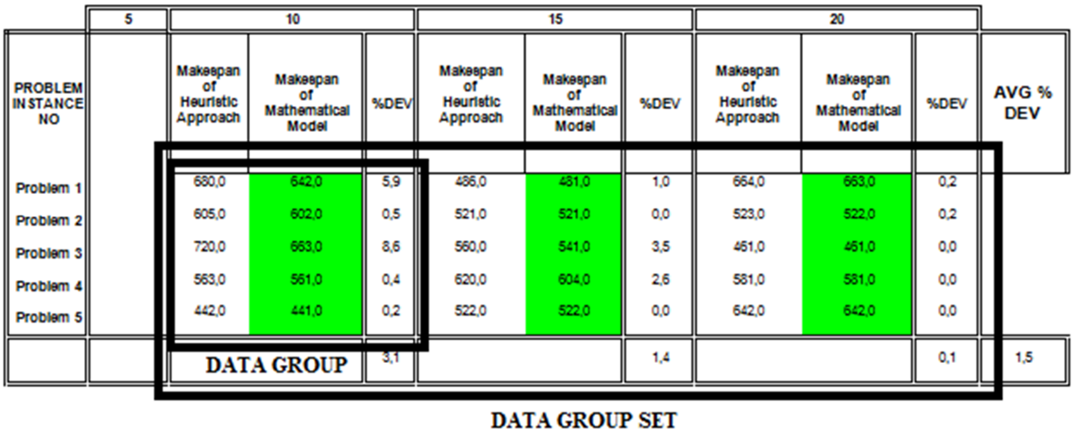
This chapter begins with the brief explanation of comparative computational results of the mathematical model and the heuristic algorithm for equal sublot case. Then we switch our focus on unequal sublot case and again we explain the comparative results.

For analyzing the performance of the mathematical model and heuristic algorithm, we created different MPLS problems on two machines. We use 5, 10, 15, 20 and 25 number of jobs and for each job we assume total number of sublots as twofold, threefold and fourfold of total number of jobs. As represented in Figure 6.1 and Figure 6.2, for each job-sublot pair, we create five samples. In the following sections, we call *data group* to these five-sampled MPLS problems. Also we called *data group set* to three-itemed set of data groups which each data group has same number of jobs but different total number of sublots.



PROBLEM INSTANCE NO	NUMBER OF JOB	NUMBER OF SUBLOTS								
		50			75			100		
		Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1	25	1321,182	1325	1322,35	1976,098	1976,8	1979,25	1635,029	1641,175	1661,25
Problem 2		1420,344	1421,52	1422,4	2016,196	2016,4	2017,677	1427,159	1428,293	1434,111
Problem 3		1572,144	1575,6	1572,5	1445,046	1452	1448,5	1856,174	1863,925	1860,615
Problem 4		1236,984	1239,4	1238	1457,158	1467	1461	1924,174	1925,44	1996,235
Problem 5		1436,958	1437	1433,5	1383,902	1388	1386	1772,04	1772,201	1776,5

Figure 6.1 Data Group and Data Group Set Representations for Unequal Sized Sublots Case



PROBLEM INSTANCE NO	5	10			15			20			AVG % DEV
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		680,0	642,0	5,9	486,0	481,0	1,0	664,0	663,0	0,2	
Problem 2		605,0	602,0	0,5	521,0	521,0	0,0	523,0	522,0	0,2	
Problem 3		720,0	663,0	8,6	560,0	541,0	3,5	461,0	461,0	0,0	
Problem 4		563,0	561,0	0,4	620,0	604,0	2,6	581,0	581,0	0,0	
Problem 5		442,0	441,0	0,2	522,0	522,0	0,0	642,0	642,0	0,0	
				3,1			1,4			0,1	1,5

Figure 6.2 Data Group and Data Group Set Representations for Equal Sized Sublots Case

For qualifying the solution success, we compare the makespan values obtained by the heuristic algorithm and the mathematical model with respect to changing total lot size, sublot size etc., and we analyze the behavior of the heuristic algorithm and the mathematical model for each data groups. We present makespan values in tabular form as seen on Figure 6.1 and Figure 6.2 and, we will use color code; *green* for normally completed mathematical model and *yellow* for the mathematical model that throws time limit or resource limit exceeded error. So for the green ones we except that makespan of mathematical model obtained by GAMS satisfy the optimal solution and for the yellow ones makespan of mathematical model obtained by GAMS is near optimal.

We create our problem instances in three different scenarios. For the first scenario, for each job-sublot pair, we assume total lot size is 20 and by changing the processing time for one unit of job for each pair, we create totally 75 different MPLS problem instances. For the second scenario, we use the same parameters of first scenario except the total lot size. We randomly generate the total lot size between 2 and 20 for each job-sublot pair and we compute the result of MPLS problem on 75 different problem instances. For the last scenario, we create problem instances that for each job-sublot pair, the processing time of each job on first machine is less than or equal to the processing time of each job on the second machine and we compute the result of 75 different MPLS problem. Thus, totally, 225 problem instances are used to measure the effectiveness of the mathematical models and the heuristic algorithms.

6.1 Comparative Computational Results of the Mathematical Model and the Heuristic Algorithm for the Case with Equal Sized Sublots

In this section, we compare the makespan values obtained by the mathematical model and the heuristic algorithm for MPLS problem with equal sized sublots.

Below, we compare the makespan deviation of heuristic algorithm from mathematical model with respect to changing total lot size, sublot size etc. When we choose the final makespan value of heuristic algorithm, for each instance of data, we pick the smallest average deviation from makespan of Step 1 and Step 2 of heuristic approach's solutions.

For example, assume we have the following makespan values: Table 6-1 for Data Group 1; Table 6-2 for Data Group 2 and Table 6-3 for Data Group 3.

Table 6-1 Number of Jobs is 5 and Number of Sublots is 10

	Step 1 of Heuristic Approach Makespan of Split All Lots of Jobs Equally and Pick Smallest Heuristic Approach	Step 2 of Heuristic Approach Makespan of Split Two Jobs with the First and Second Highest Processing Times Heuristic Approach	Makespan of Mathematical Model	% Deviation between Step 1 of Heuristic Approach Makespan and Mathematical Mode Makespanl	% Deviation between Step 2 of Heuristic Approach Makespan and Mathematical Model Makespan	% Deviation between Mathematical Modal Makespan and Best of Step1 Makespan and Step 2 Makespan
Problem 1	265,481	268,6	265,5	0,0	1,2	0,0
Problem 2	480,35	491	480,333333	0,0	2,2	0,0
Problem 3	327,01	327	311,8	4,9	4,9	4,9
Problem 4	402	402,6	402	0,0	0,1	0,0
Problem 5	220,996	219	215	2,8	1,9	1,9
				Average:1,5	Average:2,1	Average:1,3

Table 6-2 Number of Jobs is 5 and Number of Sublots is 15

	Step 1 of Heuristic Approach Makespan of Split All Lots of Jobs Equally and Pick Smallest Heuristic Approach	Step 2 of Heuristic Approach Makespan of Split Two Jobs with the First and Second Highest Processing Times Heuristic Approach	Makespan of Mathematical Model	% Deviation between Step 1 of Heuristic Approach Makespan and Mathematical Mode Makespanl	% Deviation between Step 2 of Heuristic Approach Makespan and Mathematical Model Makespan	% Deviation between Mathematical Modal Makespan and Best of Step1 Makespan and Step 2 Makespan
Problem 1	329,23	330,2	322,222222	2,2	2,5	2,2
Problem 2	349,285	349,9	349,272727	0,0	0,2	0,0
Problem 3	433,525	434	433,545455	0,0	0,1	0,0
Problem 4	438,982	409	393	11,7	4,1	4,1
Problem 5	291,98	292	283,111111	3,1	3,1	3,1
				Average:3,4	Average:2,0	Average:1,9

Table 6-3 Number of Jobs is 5 and Number of Sublots is 20

	Step 1 of Heuristic Approach Makespan of Split All Lots of Jobs Equally and Pick Smallest Heuristic Approach	Step 2 of Heuristic Approach Makespan of Split Two Jobs with the First and Second Highest Processing Times Heuristic Approach	Makespan of Mathematical Model	% Deviation between Step 1 of Heuristic Approach Makespan and Mathematical Mode Makespanl	% Deviation between Step 2 of Heuristic Approach Makespan and Mathematical Model Makespan	% Deviation between Mathematical Modal Makespan and Best of Step1 Makespan and Step 2 Makespan
Problem 1	534	534	525,2	1,7	1,7	1,7
Problem 2	286,5	302,02	285,285714	0,4	5,9	0,4
Problem 3	346,006	346,098	346,062501	0,0	0,0	0,0
Problem 4	338,625	341,841	338,625	0,0	0,9	0,0
Problem 5	340	339,97	327	4,0	4,0	4,0
				Average:1,2	Average:2,5	Average:1,2

In order to analyze performance of heuristic algorithm and mathematical model, we calculate the deviation of heuristic algorithm's makespan from mathematical model's makespan. As shown in Table 6-1, Table 6-2 and Table 6-3, the makespan value of algorithm and heuristic approach increase when number of sublot is increased, as expected. But the average percentage deviations of heuristic algorithm from mathematical model are not increased linearly when numbers of sublots are increased. Also heuristic algorithm's Step1 provides 33% smaller average percentage makespan deviation than heuristic algorithm's Step 2 does. Also, 46.6% of these problem instances, heuristic algorithm's makespan value equals to mathematical model's makespan value. It means that 46.6% of these three data groups, heuristic algorithm provide optimal solution.

Now, as we demonstrated above, we analyze comparative computational results of mathematical model and heuristic algorithm for our three different scenarios.

For the first scenario, as expected, when number of jobs and number of sublots are increased, the makespan of heuristic algorithm and mathematical model are increased. The details of makespan values for changing total number of jobs are given in Table 6-4, Table 6-5, Table 6-6, Table 6-7 and Table 6-8. As shown on these tables, when number of jobs is increased, mathematical model throws time limit or resource limit exceeded warn more often. When number of job is 5, two problem instances provides near optimal solution ;but when number of job is 10 only one problem instance provides near optimal solution and for 15, 20 and 25 numbers of jobs all problem instances provides near optimal solution. That means when number of jobs are increased, mathematical model provides the near optimal solution, not optimal solution.

Table 6-4 Makespan values for 5 Number of Jobs for Scenario 1

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		10			20			30		
		Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1	5	646,648	646,648	646,666667	500,0	492,0	482,5	665	669,286	664
Problem 2		606,65	620	606,666667	521,8	522,0	521,818182	523,75	525,287	522,666667
Problem 3		669,983	690	670	541,8	542,0	541,818182	461,25	479,99	461,250001
Problem 4		563,317	570	563,333333	680,0	620,0	611,428571	581,25	583,959	581,333335
Problem 5		443,319	448	443,333333	523,6	524,0	523,636364	642,5	659,97	642,5

Table 6-5 Makespan values for 10 Number of Jobs for Scenario 1

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		20			30			40		
		Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1	10	1001,812	1010	1002,00008	880,9	884,0	881,25	1081,255	1100,08	1081,935484
Problem 2		1021,8	1022	1021,818184	861,8	862,0	862,500001	1100,605	1100,747	1100,869565
Problem 3		1221,8	1222	1221,818215	1120,9	1121,0	1121,111117	981,25	981,414	981,600013
Problem 4		1323,626	1332	1322,500032	1000,9	1003,0	1001	1300,605	1302,091	1300,740742
Problem 5		1261,802	1270	1261,818182	1360,9	1361,0	1361,052654	1200,625	1201,414	1201,111111

Table 6-6 Makespan values for 15 Number of Jobs for Scenario 1

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		30			45			60		
		Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1	15	1261,25	1261,288	1261,333334	1580,6	1582,8	1581,052632	1880,91	1880,708	1885
Problem 2		1801,25	1801,288	1801,333333	1480,6	1482,1	1482,5	1920,465	1920,264	1920,555556
Problem 3		1621,25	1623,456	1621,428942	1780,6	1782,8	1781,052632	1880,485	1880,708	1883,333333
Problem 4		1701,25	1701,288	1701,428588	1660,6	1660,8	1660,689685	1300,92	1301,596	1308
Problem 5		1421,25	1429,97	1421,333333	1680,6	1682,8	1680,740743	1620,92	1620,708	1624

Table 6-7 Makespan values for 20 Number of Jobs for Scenario 1

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		40			60			80		
		Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1	20	2320,888	2324	2322,000013	1980,5	1981,0	1982,352941	2040,36	2040,153	2043,52381
Problem 2		1800,88	1801	1801,666677	1980,5	1981,5	1980,714286	2300,344	2300,153	2306,666667
Problem 3		2100,888	2120	2104,000149	1860,5	1860,5	1860,833333	1840,344	1840,153	1840,327869
Problem 4		2140,896	2142	2140,952381	2060,6	2060,5	2068,888889	2180,352	2180,153	2188,571432
Problem 5		2420,896	2422	2422,222223	2200,5	2201,5	2205,454547	2100,368	2101,485	2105,000001

Table 6-8 Makespan values for 25 Number of Jobs for Scenario 1

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		50			75			100		
	25	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		2580,727	2600	2582,000005	3020,4	3020,8	3024	2840,191	2841,293	2965,500394
Problem 2		2580,739	2581,6	2586,666667	3020,3	3020,4	3021,904766	2520,155	2520,492	2599,393939
Problem 3		2540,715	2540,8	2544	2340,3	2341,6	2342,222222	2900,179	2900,759	2967,681159
Problem 4		2639,946	2640	2628	2480,3	2480,8	2490	2960,167	2961,827	3150,965517
Problem 5		2640,739	2641,6	2650,000001	2300,3	2320,0	2300,784314	2800,191	2800,759	2860

We start first scenario's analysis by analyzing the performance of heuristic algorithm. If we look at Table 6-9, for only one data group, Step2 of heuristic algorithm provides smaller average percentage makespan deviation than Step1 provides. So when we compare the average percentage makespan deviation of heuristic algorithm from mathematical model, we often use makespan value of heuristic algorithm's Step2 as heuristic algorithm's makespan value.

If we analyze the performance of heuristic algorithm with respect to mathematical model, we have to compare their success to reach optimal makespan values. Below in Table 6-9, a blue bannered makespan deviation highlights the makespan values that heuristic algorithm provides smaller makespan than mathematical model does.

Table 6-9 The Deviation of Heuristic Approaches from Mathematical Model Solved by GAMS with respect to Changing Number of Jobs and Total Lot Sizes for the First Scenario

NUMBER OF JOBS	NUMBER OF SUBLOTS	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
5	10	0,0	1,5
5	15	3,0	0,7
5	20	0,1	1,7
10	20	0,0	0,4
10	30	0,0	0,1
10	40	0,0	0,4
15	30	0,0	0,1
15	45	0,0	0,1
15	60	-0,2	-0,2
20	40	-0,1	0,2
20	60	-0,1	-0,1
20	80	-0,2	-0,2
25	50	-0,1	0,1
25	75	-0,1	0,1
25	100	-3,5	-3,5

As shown in Figure 6.3, when total number of jobs and total number of sublots are increased, heuristic algorithm provides smaller makespan values than mathematical model provides. If we look at Figure 6.3, when total number of jobs increases, the average percentage makespan deviation of heuristic algorithm from mathematical model decreases, even more it goes negative values. When total number of jobs increases, mathematical model is unable to reach optimal makespan even heuristic algorithm achieves it. If we analyze the average percentage makespan deviation values, as seen on Figure 6.3, average percentage makespan deviation of heuristic is 0.26% max; -1.23% min and -0.2% on average.

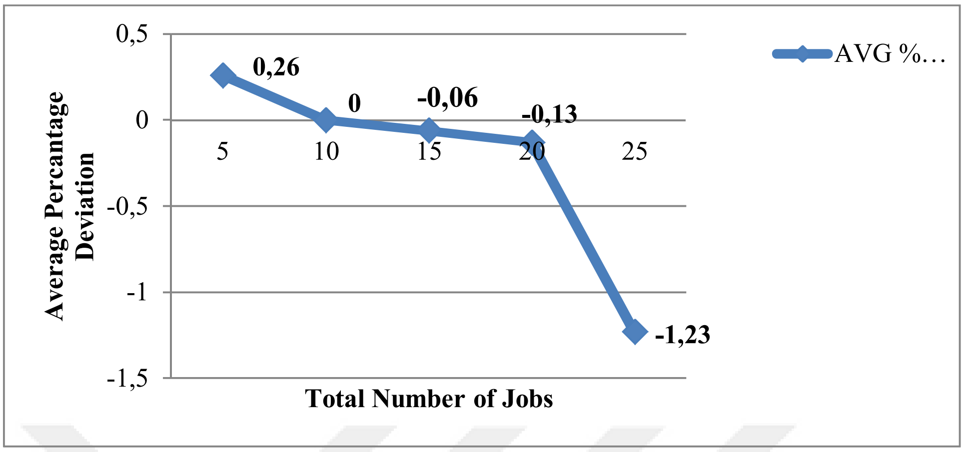


Figure 6.3 Average Percentage Deviations for Changing Number of Jobs for Scenario 1

For the second scenario, as expected, when number of jobs and number of sublots are increased, the makespan of heuristic algorithm and mathematical model are increased. Also, it is important to point out when number of jobs is increased, mathematical model throws time limit or resource limit exceeded warn more often. The details of makespan values for changing total number of jobs are given in Table 6-10, Table 6-11, Table 6-12, Table 6-13 and Table 6-14.

As shown in these tables, when number of job is 5, only one problem instance provides near optimal solution; when job number is 10, 15, 20 and 25, all sixty solutions are near optimal. That means when number of jobs are increased, mathematical model provides the near optimal solution, not optimal solution.

Table 6-10 Makespan values for 5 Number of Jobs for Scenario 2

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		10			15			20		
	5	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		265,481	268,6	265,5	329,23	330,2	322,2222	534	534	525,2
Problem 2		480,35	491	480,3333	349,285	349,9	349,2727	286,5	302,02	285,2857
Problem 3		327,01	327	311,8	433,525	434	433,5455	346,006	346,098	346,0625
Problem 4		402	402,6	402	438,982	409	393	338,625	341,841	338,625
Problem 5		220,996	219	215	291,98	292	283,1111	340	339,97	327

Table 6-11 Makespan values for 10 Number of Jobs for Scenario 2

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		20			30			40		
	10	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		597,985	598	588,5715	665,75	665,85	664,1765	562,21	563,809	562,6429
Problem 2		708,9	717,5	709	440,98	444,55	440,6668	839,994	855,08	839,5714
Problem 3		642,976	643	637,75	839,84	840	840,1111	550	550,214	549,5333
Problem 4		699,814	704,4	699,1111	497,61	498,95	497,6667	481,25	489,03	481,5714
Problem 5		936,505	942	936,6	555,27	561	555,3	726,375	727,186	726,4483

Table 6-12 Makespan values for 15 Number of Jobs for Scenario 2

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		30			45			60		
	15	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		780,625	781	781,25	996,395	998,442	996,3667	1195,232	1198,2	1195,737
Problem 2		1091,498	1094,954	1091,563	898,107	899,809	898,2174	1297,186	1304,86	1297,195
Problem 3		1076,084	1084,475	1076,357	1034,22	1037,492	1034,6	1164,03	1169,88	1164,462
Problem 4		1073,375	1079,045	1073,4	1092,342	1092,757	1092,333	786,368	787,199	790,4444
Problem 5		869,625	874,97	869,4286	1120,342	1130,08	1120,345	1039,308	1040,608	1039,833

Table 6-13 Makespan values for 20 Number of Jobs for Scenario 2

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		40			60			80		
	20	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		1327,556	1332,7	1328	1183,905	1184	1179,467	1285	1287,1	1287,252
Problem 2		1127,88	1128	1127,952	1293,224	1294,275	1293,333	1500,226	1513,9	1504,1
Problem 3		1307,996	1308	1304,455	1111,14	1116	1111,778	1500,226	1513,9	1504,1
Problem 4		1251,458	1260	1251,9	1142,27	1149	1142,583	1178,032	1183,9	1180,909
Problem 5		1381,232	1386	1381,421	1284,16	1292	1285,854	1296,988	1298,838	1297,919

Table 6-14 Makespan values for 25 Number of Jobs for Scenario 2

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		50			75			100		
	25	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		1321,182	1325	1322,35	1976,098	1976,8	1979,25	1635,029	1641,175	1661,25
Problem 2		1420,344	1421,52	1422,4	2016,196	2016,4	2017,677	1427,159	1428,293	1434,111
Problem 3		1572,144	1575,6	1572,5	1445,046	1452	1448,5	1856,174	1863,925	1860,615
Problem 4		1236,984	1239,4	1238	1457,158	1467	1461	1924,174	1925,44	1996,235
Problem 5		1436,958	1437	1433,5	1383,902	1388	1386	1772,04	1772,201	1776,5

Also, it is important to point that, for

- Number of jobs 10; 2 problem instances on 40 number of sublots
- Number of sublots 15; 1 problem instances on 15 number of sublots and 2 problem instances on 60 number of sublots

- Number of sublots 20; 2 problem instances on 60 number of sublots and 5 problem instances on 80 number of sublots
- Number of sublots 25; 3 problem instances on 50 number of sublots; 5 problem instances on 75 number of sublots and 5 problem instances on 100 number of sublots

heuristic algorithm provides 0.1% average smaller makespan value than mathematical model does. Table 6-15 presents the average percentage makespan deviation of heuristic algorithm from mathematical model with respect to changing number of jobs and total lot sizes. Blue bannered makespan deviations show that, on average, heuristic algorithm provides smaller makespan than mathematical model provides for that data group. According to results, it is obvious that when number of sublots is increased, mathematical model is unable to reach optimal solution even heuristic algorithm reaches it.

Table 6-15 The Deviation of Heuristic Approaches from Mathematical Model Solved by GAMS with respect to Changing Number of Jobs and Total Lot Sizes for the Second Scenario

NUMBER OF JOBS	NUMBER OF SUBLOTS					
5	10		15		20	
	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	1,5	2,1	3,4	2,0	1,2	2,5
10	20		30		40	
	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	0,5	1,0	0,1	0,5	-0,1	0,8
15	30		45		60	
	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	0,0	0,4	0,0	0,3	-0,1	0,3
20	40		60		80	
	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	0,0	0,3	0,0	0,4	10,8	0,3
25	50		75		100	
	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	0,0	0,2	-0,1	0,2	-0,1	-0,1

If we analyze behavior of average percentage makespan deviation of heuristic algorithm from mathematical model, Figure 6.4 shows that when number of jobs is 5, average percentage makespan deviation is approximately 1.56% but when total number of jobs is 25 average percentage makespan deviations is approximately -0, 66%. So it is obvious that when total number of jobs is increased, average percentage makespan deviation of heuristic algorithm from mathematical model is decreased. It means when total number of jobs is increased, heuristic algorithm provides better makespan for this scenario. Moreover, if we analyze the performance of heuristic algorithm when total number of jobs is fixed and total number of sublots changed, we don't reach as certain conclusion as previous sample. If we look at Table 6-15, when total number of jobs is fixed and number of sublots is increased, the average percentage makespan deviation of heuristic algorithm from mathematical model neither continuously increasing, nor continuous decreasing. It behaves randomly for that case for this scenario.

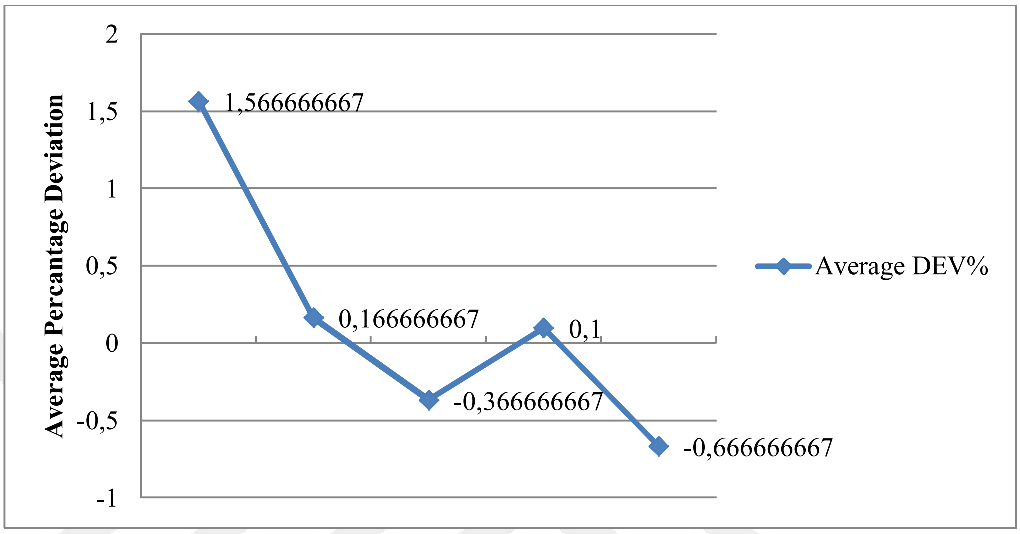


Figure 6.4 Average Percentage Deviations for Changing Number of Jobs for Scenario 2

For the third scenario, as for the second scenario, when number of jobs and number of sublots are increased, the makespan of heuristic algorithm and mathematical model are increased. The details of makespan values for changing total number of jobs are given in Table 6-16, Table 6-17, Table 6-18, Table 6-19 and Table 6-20. Again as in the previous scenarios, normal completed mathematical model's makespan times are represented as green and time limit or resource limit exceeded makespan times are represented in yellow. It is obvious that when number of jobs is increased, mathematical model fails to reach optimal solution. Also, the same results are received when number of jobs is fixed and number of total sublots is increased. As shown in Table 6-21, when total number of sublots is threefold of total number of jobs approximately 40% of mathematical models are unable to provide optimal solution.

Table 6-16 Makespan values for 5 Number of Jobs for Scenario 3

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		10			15			20		
	5	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		265	270	265	363,154	366,4	363,1818	367,125	367,2	367,125
Problem 2		282,648	284	279,2	296,44	299,7	294,7	307,076	308,641	306,3333
Problem 3		241,314	244	237,6	436,465	439	436,4545	466	466,288	466
Problem 4		228,014	230	220,5	203,901	204	203,5	315,875	319,098	314,5333
Problem 5		264	257	256	297,992	299,5	296,4	305	304,955	301,4444

Table 6-17 Makespan values for 10 Number of Jobs for Scenario 3

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		20			30			40		
	10	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		499	506	498,8889	719,232	721,85	719,3333	394,27	397	394,5455
Problem 2		581,252	590,8	581,3333	551,688	552,5	551,7778	630,397	631,809	630,25
Problem 3		679,564	682	677,5556	826,681	832	826,9333	538,342	540,585	538,4074
Problem 4		524,47	529	524,5556	459,386	459,6	459,4706	624,44	628,2	624,3333
Problem 5		368,39	372,1	368	450,27	452,2	450,2857	495,349	497,518	495,3846

Table 6-18 Makespan values for 15 Number of Jobs for Scenario 3

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		30			45			60		
	15	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		889,625	894,38	889,8	675,375	676,206	675,4	696,07	698,4	696,1579
Problem 2		735,25	739	735,4001	897,254	899,472	897,5	1068,197	1069,192	1068,308
Problem 3		1191,28	1196,04	1191,5	880,256	881,5	880,2593	1162,97	1164,602	1163,294
Problem 4		936,5	942,04	936,5455	874,12	876,061	874,2	765,096	767,1	765,1944
Problem 5		938	938,826	938,2	1195,73	1199,341	1195,429	827,181	827,329	827,6842

Table 6-19 Makespan values for 20 Number of Jobs for Scenario 3

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		40			60			80		
	20	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		1085,319	1086,5	1085,429	1526,189	1529	1526,366	1204,13	1204,859	1205,5
Problem 2		1123,3	1127,75	1123,444	1268,244	1270,375	1268,7	1491,14	1491,506	1493,273
Problem 3		1384,616	1390	1384,813	1357,303	1357,45	1357,818	1155,922	1157,218	1157
Problem 4		1178,232	1179,9	1178,294	1464,185	1466,25	1464,308	1112,042	1113,048	1113,444
Problem 5		1312,1	1315,6	1312,25	933,062	935,85	933,1818	906,9	907,25	907,6667

Table 6-20 Makespan values for 25 Number of Jobs for Scenario 3

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS								
		50			75			100		
	25	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model	Makespan of Heuristic Approach's Step1	Makespan of Heuristic Approach's Step2	Makespan of Mathematical Model
Problem 1		1637,136	1641,8	1637,263	1711,059	1711,8	1711,4	1267,045	1268,4	1269,381
Problem 2		1282,168	1285,2	1282,56	1564,171	1565,14	1567	1520,072	1521,802	1526,875
Problem 3		1573,174	1577	1574,4	1658,059	1659,44	1659,036	1815,029	1816,04	1823
Problem 4		1059,168	1061,56	1059,385	1490,176	1491,8	1490,5	1548,965	1549,96	1551,667
Problem 5		1561,322	1566,6	1561,375	1602,175	1604,28	1602,682	1529,029	1529,881	1531,082

As seen on Table 6-21, when number of total number of jobs increases, the average percentage makespan deviation of heuristic algorithm from mathematical model decreases, almost they are same, 0%. It shows that when total number of jobs and total number of sublots increase, average percentage makespan deviations are close to each other, as well they are same. Also, number of blue bannered makespan values is increased when total number of jobs is increased as in the second scenario, but this time deviation is negligible.

Table 6-21 The Deviation of Heuristic Approaches from Mathematical Model Solved by GAMS with respect to Changing Number of Jobs and Total Lot Sizes for the Third Scenario

NUMBER OF JOBS	NUMBER OF SUBLOTS					
	10		15		20	
5	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	1,9	2,2	0,3	0,9	0,4	0,7
	20		30		40	
10	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	0,1	1,1	0,0	0,3	0,0	0,5
	30		45		60	
15	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	0,0	0,4	0,0	0,2	0,0	0,1
	40		60		80	
20	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	0,0	0,2	0,0	0,1	-0,1	0,0
	50		75		100	
25	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2	DEV% of Heuristic Approach's Step1	DEV% of Heuristic Approach's Step2
	0,0	0,2	0,0	0,0	-0,1	0,0

6.2 Comparative Computational Results of the Mathematical Model and the Heuristic Algorithm for the Case with Unequal Sized Sublots

In this section, we compare the solutions obtained by the mathematical model and the heuristic algorithm for MPLS problem with unequal sized sublots.

For the first scenario, as expected, when the number of jobs and the number of sublots are increased, the makespan values obtained by heuristic algorithm and the mathematical model are increased as shown in Table 6-22. Also Table 6-22 shows that mathematical model provides the optimal makespan value for all problem instances.

If we analyze the average percentage makespan deviation for changing total number of jobs, as you seen in Figure 6.5, the average percentage makespan deviation of heuristic algorithm from mathematical model is 1.5% maximum, 0% minimum and 0.38% on average. We conclude that Scenario1's heuristic approach provides very close solution to optimal solution.

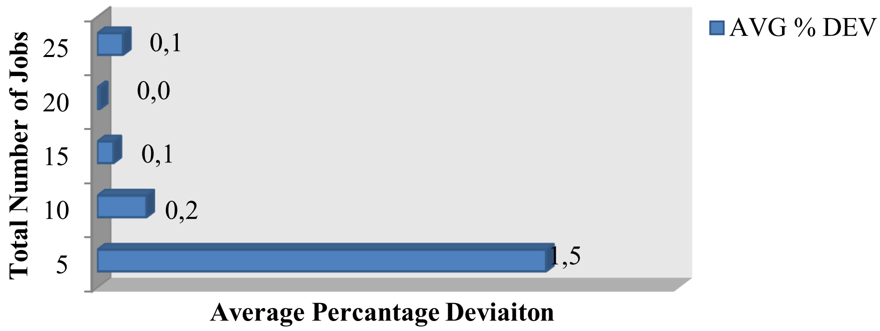


Figure 6.5 Makespan Deviation's Average Percentage for each Total Number of Jobs for Scenario 1

Moreover, if we analyze Table 6-22, when total number of jobs is 10, 15, 20 and 25 average percentage makespan deviation of data group sets is 0.1%, but when total number of jobs is 5, average percentage makespan deviation of data group set is 1.5%. So we conclude that when total number of jobs is increased, average percentage makespan deviation of heuristic algorithm from mathematical model decreases, mostly. But we can't conclude that when total number of sublots is increased, the average makespan percentage deviation doesn't always increases or decreases for all data group sets. For example for data group set with total number of jobs is 5, when number of sublots is increased from two fold to three fold of number of jobs, average percentage makespan deviation decreases but when number of sublots is increased from three fold to fourfold of number of jobs; average percentage makespan deviation decreases. On the other hand, for data group set with total number of jobs is 20, when number of sublots is increased from two fold, three fold or four fold of number of jobs; average percentage makespan deviation is same, 0%. So it is concluded that when total number of jobs is fixed and total number of sublots are increased as a multiple of total number of jobs, the average makespan percentage deviation neither decreases, nor increases; behaves unpredictably.

Table 6-22 Scenario 1’s Makespan Values for MPLS Problem with Unequal Sublot

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS									AVG % DEV
	5	10			15			20			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		680,0	642,0	5,9	486,0	481,0	1,0	664,0	663,0	0,2	
Problem 2		605,0	602,0	0,5	521,0	521,0	0,0	523,0	522,0	0,2	
Problem 3		720,0	663,0	8,6	560,0	541,0	3,5	461,0	461,0	0,0	
Problem 4		563,0	561,0	0,4	620,0	604,0	2,6	581,0	581,0	0,0	
Problem 5		442,0	441,0	0,2	522,0	522,0	0,0	642,0	642,0	0,0	
				3,1			1,4			0,1	1,5
	10	20			30			40			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		1005,0	1001,0	0,4	881,0	881,0	0,0	1083,0	1082,0	0,1	
Problem 2		1021,0	1021,0	0,0	862,0	862,0	0,0	1101,0	1101,0	0,0	
Problem 3		1221,0	1221,0	0,0	1121,0	1121,0	0,0	982,0	982,0	0,0	
Problem 4		1332,0	1321,0	0,8	1002,0	1001,0	0,1	1303,0	1301,0	0,2	
Problem 5		1272,0	1261,0	0,9	1361,0	1361,0	0,0	1202,0	1201,0	0,1	
AVG DEV				0,4			0,0			0,1	0,2
	15	30			45			60			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		1261,0	1261,0	0,0	1581,0	1581,0	0,0	1882,0	1882,0	0,0	
Problem 2		1803,0	1801,0	0,1	1483,0	1481,0	0,1	1921,0	1921,0	0,0	
Problem 3		1621,0	1621,0	0,0	1786,0	1781,0	0,3	1882,0	1881,0	0,1	
Problem 4		1701,0	1701,0	0,0	1661,0	1661,0	0,0	1302,0	1301,0	0,1	
Problem 5		1421,0	1421,0	0,0	1684,0	1681,0	0,2	1622,0	1622,0	0,0	
AVG DEV				0,0			0,1			0,0	0,1
	20	40			60			80			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		2324,0	2321,0	0,1	1981,0	1981,0	0,0	2041,0	2041,0	0,0	
Problem 2		1801,0	1801,0	0,0	1981,0	1981,0	0,0	2301,0	2301,0	0,0	
Problem 3		2101,0	2101,0	0,0	1861,0	1861,0	0,0	1841,0	1841,0	0,0	
Problem 4		2142,0	2141,0	0,0	2061,0	2061,0	0,0	2181,0	2181,0	0,0	
Problem 5		2420,0	2421,0	0,0	2201,0	2201,0	0,0	2101,0	2101,0	0,0	
AVG DEV				0,0			0			0,0	0,0
	25	50			75			100			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		2587,0	2581,0	0,2	3022,0	3021,0	0,0	2844,0	2841,0	0,1	
Problem 2		2582,0	2581,0	0,0	3021,0	3021,0	0,0	2521,0	2521,0	0,0	
Problem 3		2541,0	2541,0	0,0	2341,0	2341,0	0,0	2901,0	2901,0	0,0	
Problem 4		2640,0	2621,0	0,7	2482,0	2481,0	0,0	2961,0	2961,0	0,0	
Problem 5		2644,0	2641,0	0,1	2301,0	2301,0	0,0	2802,0	2801,0	0,0	
AVG DEV				0,2			0,0			0,0	0,1
AVG % DEV				0,8			0,3			0,0	

For the second scenario, as expected, when number of jobs and number of sublots are increased, the makespan of heuristic algorithm and mathematical model are increased as shown in Table 6-23. Here, it is important to notice that only one mathematical model throws the time limit or resource limit exceeded warn and it is highlighted in yellow in Table 6-23. So except that problem instance, we presume that mathematical model's makespan is optimal. Also we conclude that heuristic algorithm provides very close solution to optimal solution. If we look at Figure 6.7, average percentage makespan deviation of heuristic algorithm from mathematical model is 1.1% maximum, 0% minimum and 0.5% on average. So for Scenario2, we conclude that heuristic algorithm provides very close solution to optimal solution.

Moreover, if we look through the average percentage makespan deviation of heuristic algorithm from mathematical model, in Table 6-23, we conclude that when number of job is fixed and number of sublots is increased by multiple of number of jobs, the behavior of average percentage deviation is not always increased or decreased. Average percentage deviation's behavior is unpredictable.

Table 6-23 Scenario 2’s Makespan Values for MPLS Problem with Unequal Sublot

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS									AVG % DEV
	5	10			15			20			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		265,0	265,0	0,0	326,0	321,0	1,6	529,0	525,0	0,8	
Problem 2		480,0	477,0	0,6	349,0	349,0	0,0	286,0	286,0	0,0	
Problem 3		322,0	310,0	3,9	433,0	433,0	0,0	346,0	346,0	0,0	
Problem 4		400,0	400,0	0,0	409,0	389,0	5,1	339,0	339,0	0,0	
Problem 5		214,0	214,0	0,0	292,0	284,0	2,8	330,0	328,0	0,6	
				0,9			1,9			0,3	1,0
	10	20			30			40			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		592,0	588,0	0,7	665,0	665,0	0,0	563,0	563,0	0,0	
Problem 2		709,0	709,0	0,0	443,0	442,0	0,2	855,0	840,0	1,8	
Problem 3		643,0	638,0	0,8	840,0	840,0	0,0	551,0	551,0	0,0	
Problem 4		701,0	699,0	0,3	498,0	498,0	0,0	484,0	483,0	0,2	
Problem 5		942,0	937,0	0,5	556,0	556,0	0,0	728,0	727,0	0,1	
AVG DEV				0,5			0,0			0,4	0,3
	15	30			45			60			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		781,0	781,0	0,0	997,0	997,0	0,0	1198,0	1041,0	15,1	
Problem 2		1092,0	1092,0	0,0	901,0	899,0	0,2	1298,0	1298,0	0,0	
Problem 3		1076,0	1076,0	0,0	1036,0	1035,0	0,1	1166,0	1165,0	0,1	
Problem 4		1077,0	1074,0	0,3	1093,0	1093,0	0,0	788,0	787,0	0,1	
Problem 5		870,0	870,0	0,0	1121,0	1121,0	0,0	1043,0	1041,0	0,2	
AVG DEV				0,1			0,1			3,1	1,1
	20	40			60			80			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		1332,0	1328,0	0,3	1175,0	1175,0	0,0	1286,0	1286,0	0,0	
Problem 2		1128,0	1128,0	0,0	1294,0	1294,0	0,0	1501,0	1501,0	0,0	
Problem 3		1308,0	1304,0	0,3	1113,0	1112,0	0,1	972,0	972,0	0,0	
Problem 4		1252,0	1252,0	0,0	1145,0	1143,0	0,2	1179,0	1179,0	0,0	
Problem 5		1386,0	1382,0	0,3	1285,0	1285,0	0,0	1298,0	1298,0	0,0	
AVG DEV				0,2			0,052981237			0,0	0,1
	25	50			75			100			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		1322,0	1322,0	0,0	1978,0	1977,0	0,1	1636,0	1636,0	0,0	
Problem 2		1422,0	1421,0	0,1	2017,0	2017,0	0,0	1428,0	1428,0	0,0	
Problem 3		1573,0	1573,0	0,0	1447,0	1446,0	0,1	1857,0	1857,0	0,0	
Problem 4		1238,0	1237,0	0,1	1458,0	1458,0	0,0	1925,0	1925,0	0,0	
Problem 5		1435,0	1432,0	0,2	1385,0	1385,0	0,0	1773,0	1773,0	0,0	
AVG DEV				0,1			0,0			0,0	0,0
AVG % DEV				0,3			0,4			0,8	

For example if we look at the Figure 6.6, when number of jobs is 5; average percentage makespan deviation is neither increasing nor decreasing when the total number of sublots are increasing as multiple of total number of jobs. The same situation is observed for the problem instances with total number of job is 10, 15, 20 and 25 as shown in Table 6-23.

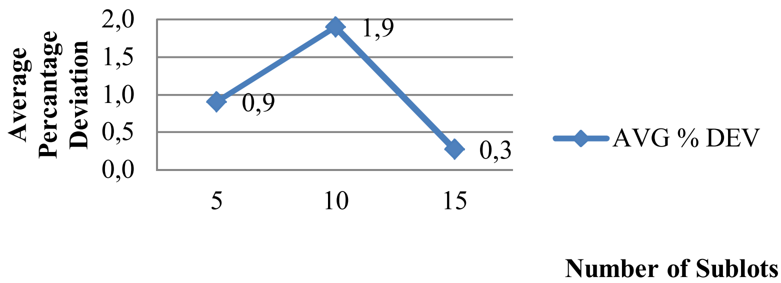


Figure 6.6 Average Percentage Deviation for Increasing Total Number of Sublots when Total Number of Jobs is 5

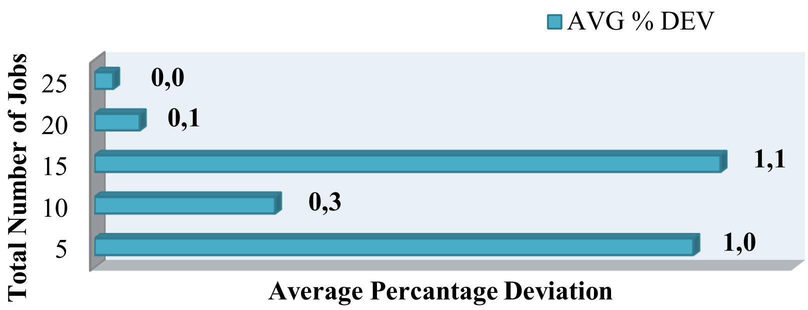


Figure 6.7 Makespan Deviation's Average Percentage for each Total Number of Jobs for Scenario 2

For the third scenario, as expected, when number of jobs and number of sublots are increased, the makespan of heuristic algorithm and mathematical model are increased as shown in Table 6-24. Also it is important to notice that all cells in are highlighted in green. It means that the mathematical model gives us the optimal solution for all problem instances in this scenario. If we analyze the average percentage makespan deviation for changing total number of jobs, as you seen in Figure 6.8, the average percentage makespan deviation of heuristic algorithm from mathematical model is 1.0% maximum, 0% minimum and 0.26% on average. So as in Scenario2, we conclude that Scenario3's heuristic algorithm provides makespan that is very close to optimal makespan.

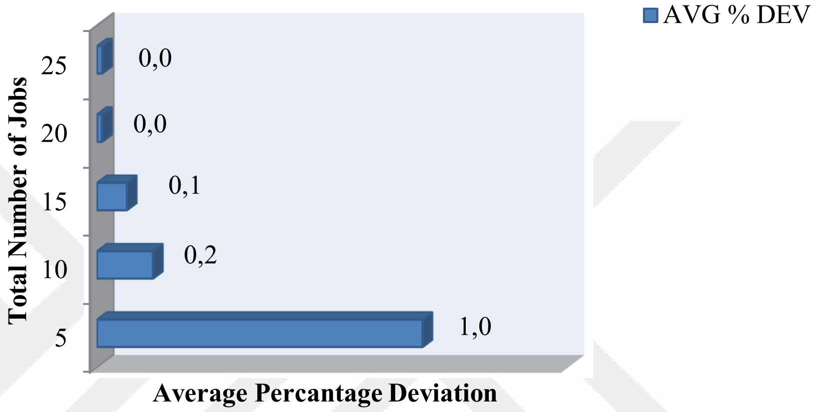


Figure 6.8 Makespan Deviation's average Percentage for each Total Number of Jobs for Scenario 3

Table 6-24 Scenario3's Makespan Values for MPLS Problem with Unequal Sublot

PROBLEM INSTANCE NO	NUMBER OF JOBS	NUMBER OF SUBLOTS									AVG % DEV
	5	10			15			20			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		270,0	265,0	1,9	363,0	363,0	0,0	367,0	367,0	0,0	
Problem 2		278,0	277,0	0,4	296,0	295,0	0,3	308,0	307,0	0,3	
Problem 3		240,0	237,0	1,3	437,0	436,0	0,2	466,0	466,0	0,0	
Problem 4		228,0	218,0	4,6	204,0	204,0	0,0	317,0	315,0	0,6	
Problem 5		264,0	250,0	5,6	296,0	296,0	0,0	303,0	302,0	0,3	
				2,7			0,1			0,3	1,0
	10	20			30			40			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		503,0	499,0	0,8	720,0	720,0	0,0	395,0	395,0	0,0	
Problem 2		582,0	581,0	0,2	553,0	552,0	0,2	631,0	631,0	0,0	
Problem 3		682,0	678,0	0,6	830,0	827,0	0,4	539,0	539,0	0,0	
Problem 4		528,0	525,0	0,6	460,0	460,0	0,0	625,0	625,0	0,0	
Problem 5		368,0	368,0	0,0	451,0	451,0	0,0	496,0	496,0	0,0	
AVG DEV				0,4			0,1			0,0	0,2
	15	30			45			60			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		890,0	890,0	0,0	676,0	676,0	0,0	697,0	697,0	0,0	
Problem 2		739,0	736,0	0,4	901,0	898,0	0,3	1069,0	1069,0	0,0	
Problem 3		1196,0	1192,0	0,3	881,0	881,0	0,0	1164,0	1164,0	0,0	
Problem 4		937,0	937,0	0,0	875,0	875,0	0,0	766,0	766,0	0,0	
Problem 5		939,0	938,0	0,1	1199,0	1196,0	0,3	828,0	828,0	0,0	
AVG DEV				0,2			0,1			0,0	0,1
	20	40			60			80			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		1087,0	1086,0	0,1	1527,0	1527,0	0,0	1205,0	1205,0	0,0	
Problem 2		1124,0	1124,0	0,0	1269,0	1269,0	0,0	1492,0	1492,0	0,0	
Problem 3		1386,0	1385,0	0,1	1358,0	1358,0	0,0	1157,0	1157,0	0,0	
Problem 4		1180,0	1179,0	0,1	1465,0	1465,0	0,0	1113,0	1113,0	0,0	
Problem 5		1313,0	1313,0	0,0	934,0	934,0	0,0	908,0	908,0	0,0	
AVG DEV				0,0			0			0,0	0,0
	25	50			75			100			
		Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	Makespan of Heuristic Approach	Makespan of Mathematical Model	%DEV	
Problem 1		1638,0	1638,0	0,0	1712,0	1712,0	0,0	1268,0	1268,0	0,0	
Problem 2		1283,0	1283,0	0,0	1566,0	1565,0	0,1	1521,0	1521,0	0,0	
Problem 3		1577,0	1574,0	0,2	1659,0	1659,0	0,0	1816,0	1816,0	0,0	
Problem 4		1060,0	1060,0	0,0	1491,0	1491,0	0,0	1550,0	1550,0	0,0	
Problem 5		1562,0	1562,0	0,0	1603,0	1603,0	0,0	1530,0	1530,0	0,0	
AVG DEV				0,0			0,0			0,0	0,0
AVG % DEV				0,7			0,1			0,1	

If we analyze the average percentage makespan deviations of heuristic algorithm from mathematical model, we conclude that when number of jobs and number of sublots are increased, the average percentage makespan deviation decreases. Below; Figure 6.9, Figure 6.10 and Figure 6.11 shows that average percentage makespan deviation of heuristic algorithm from mathematical model decreases when total number of sublots increases as multiple of number of jobs. As seen Figure 6.9, when total number of jobs is 5 and total number of sublots is one fold of number of jobs, average percentage makespan deviation is 2.7%, but when total number of jobs is 25 and total number of sublots is one fold of number of jobs average percentage makespan deviation is 0%. The same situation is observed when total number of sublots is twofold, three fold and four fold number of jobs in Figure 6.10 and Figure 6.11. So we conclude that Scenario3's average percentage makespan deviation is predictable and this value decreases when total number of jobs and total number of sublots are increased.

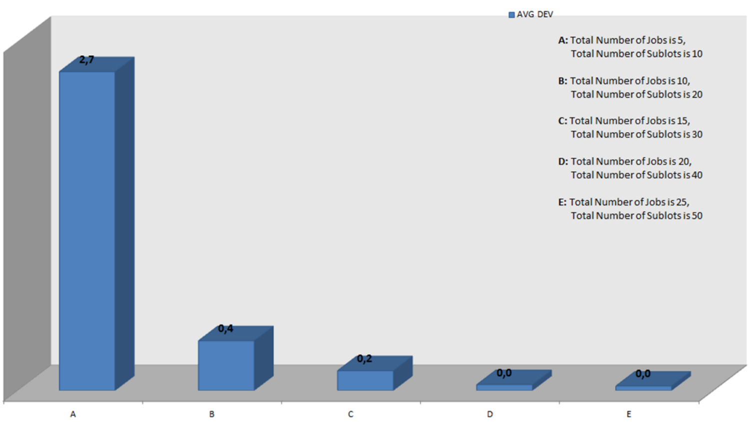


Figure 6.9 Makespan Deviation's Average Percentage when Total Number of Jobs Increases and Total Number of Sublots is Two Fold of Total Number of Jobs

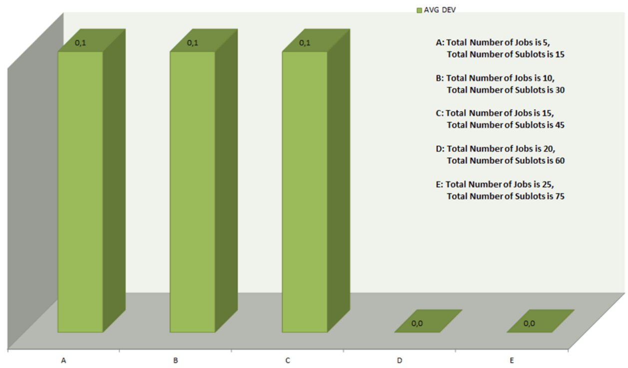


Figure 6.10 Makespan Deviation's Average Percentage when Total Number of Jobs Increases and Total Number of Sublots is Three Fold of Total Number of Jobs

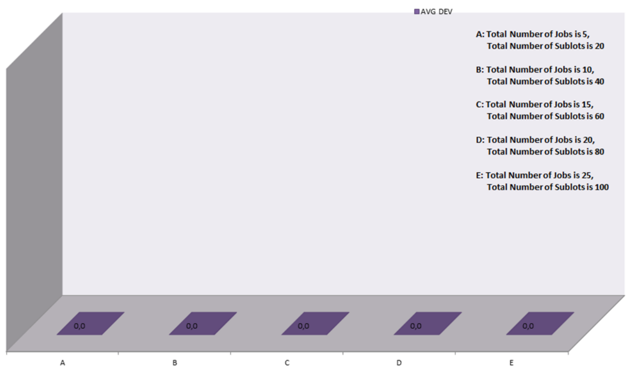


Figure 6.11 Makespan Deviation's Average Percentage when Total Number of Jobs Increases and Total Number of Sublots is Four Fold of Total Number of Jobs

CHAPTER 7

CONCLUSION

In this study, we consider a multi-product lot streaming problem on a two-machine flowshop environment in which all products are processed by Machine 1 and then by Machine 2. Most of the current studies in the literature of the multi-product lot streaming problem assume that the number of sublots for each product is known in advance, and determines the size for each sublot of every product and the sequence of sublots of all products. As opposite of the current studies in the literature, we assume that the total number of sublots for all products is known advance and our problem is to determine the number of sublots for each product, the size of each sublot and the sequence of sublots that gives the minimum makespan. We investigate the multi-product lot streaming problem for both equal and unequal sized sublots cases. For this purpose, we develop mixed integer linear mathematical models and heuristic algorithms for solving each case and compare these mathematical models with heuristic algorithms.

For unequal sublot case, the experimental studies show that almost all problem instances mathematical model provides optimal solution and thus, we were able to compare the heuristic algorithm's solution with the optimal solution. This comparison shows that heuristic algorithm provides solutions with makespan values that deviate 0.38% from the optimal solution, which is almost optimal. Also, as the total number of jobs increases, the average percent deviation of the makespan of the heuristic algorithm from the optimal makespan decreases. As the total number of jobs increases, the solution time of the MILP model by GAMS increases. However, the heuristic algorithm provides the solution in almost less than a second. Thus, for real time implementation, the solutions obtained by the heuristic algorithm can be used.

For equal sublot case, our three different scenarios showed that when the total number of jobs increases, the mathematical model cannot be solved optimally by GAMS within the allowed time limit or resource limit. However, the heuristic algorithm provides solutions in a short time.

Finally, our results of experiments show that the heuristic algorithm provides near-optimal solutions for both equal and unequal sublots cases. When we compare the solutions of the heuristic algorithms for equal and unequal sublots cases, we can easily conclude that splitting sublots unequally provides more near-optimal solutions than splitting them equally since every lot may not be split into equal sublots. For example, when the total lot size is 15 units and we split this lot into 4 equal sublots, each sublot size becomes 3.75 units and it is not possible. Only integer number sized sublots are meaningful.

Lot streaming problems with a total number of sublots for all jobs are not yet extensively studied. Thus, there is considerable number of issues remaining open for future research. Several extensions of our study can be investigated. One of them is that our problem studied in this study can be extended for more complex machining environments such as flow shops having more than two machines, jobs shops, and open shops. Study of the same problem for different performance measures such as total or maximum lateness, total completion times, and the number of tardy jobs would be some extensions.

REFERENCES

1. Defersha, F.M. and M. Chen, A hybrid genetic algorithm for flowshop lot streaming with setups and variable sublots. *International Journal of Production Research*, 2010. **48**(6): 1705-1726.
2. Kalir, A.A. and S.C. Sarin, Evaluation of the potential benefits of lot streaming in flow-shop systems. *International Journal of Production Economics*, 2000. **66**(2): 131-142.
3. Sarin, S.C. and P. Jaiprakash, Flow shop lot streaming. 2007: Springer Science & Business Media.
4. Rodoslu, E., Heuristic Approaches for the Lot Streaming Problem in Multi-Product Flow Shops. 2013(Koc University Graduate School of Sciences and Engineering): 5,13.
5. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi* 6. 131,134-135.
6. Open Shop Scheduling. Available from: https://en.wikipedia.org/wiki/Open-shop_scheduling.
7. Chiu, H.N. and J.H. Chang, Cost models for lot streaming in a multistage flow shop. *Omega*, 2005. **33**(5): 435-450.
8. Feldmann, M. and D. Biskup, On lot streaming with multiple products. 2005.
9. Chang*, J.H. and H.N. Chiu, A comprehensive review of lot streaming. *International Journal of Production Research*, 2005. **43**(8): 1515-1536.
10. Trietsch, D. and K.R. Baker, Basic techniques for lot streaming. *Operations Research*, 1993. **41**(6): 1065-1076.
11. Reiter, S., A system for managing job-shop production. *The Journal of Business*, 1966. **39**(3): 371-393.
12. Material Requirement Planning. Available from: https://en.wikipedia.org/wiki/Material_requirements_planning.
13. Manufacturing resource planning. Available from: https://en.wikipedia.org/wiki/Manufacturing_resource_planning#History_and_evolution.
14. Just in Time Manufacturing. Available from: https://en.wikipedia.org/wiki/Just-in-time_manufacturing.
15. Szendrovits, A.Z., Manufacturing cycle time determination for a multi-stage economic production quantity model. *Management Science*, 1975. **22**(3): 298-308.
16. Goyal, S., Note-Note on "Manufacturing Cycle Time Determination for a Multi-Stage Economic Production Quantity Model". *Management Science*, 1976. **23**(3): 332-333.

17. Truscott, W.G., Production scheduling with capacity-constrained transportation activities. *Journal of Operations Management*, 1986. **6**(3-4): 333-348.
18. Potts, C. and K. Baker, Flow shop scheduling with lot streaming. *Operations research letters*, 1989. **8**(6): 297-303.
19. Kropp, D.H. and T.L. Smunt, Optimal and heuristic models for lot splitting in a flow shop. *Decision sciences*, 1990. **21**(4): 691-709.
20. Baker, K. and D. Pyke, Solution Procedures for the Lot-Streaming Problem. *Decision sciences*, 1990. **21**(3): 475-491.
21. Baker, K.R. and D. Jia, A comparative study of lot streaming procedures. *Omega*, 1993. **21**(5): 561-566.
22. Glass, C., J. Gupta, and C. Potts, Lot streaming in three-stage production processes. *European Journal of Operational Research*, 1994. **75**(2): 378-394.
23. Chen, J. and G. Steiner, Lot streaming with attached setups in three-machine flow shops. *Iie Transactions*, 1998. **30**(11): 1075-1084.
24. Şen, A., E. Topaloğlu, and Ö.S. Benli, Optimal streaming of a single job in a two-stage flow shop. *European Journal of Operational Research*, 1998. **110**(1): 42-62.
25. Sriskandarajah, C. and E. Wagneur, Lot streaming and scheduling multiple products in two-machine no-wait flowshops. *IIE transactions*, 1999. **31**(8): 695-707.
26. Kumar, S., T.P. Bagchi, and C. Sriskandarajah, Lot streaming and scheduling heuristics for m-machine no-wait flowshops. *Computers & Industrial Engineering*, 2000. **38**(1): 149-172.
27. Ramasesh, R.V., H. Fu, D.K. Fong, and J.C. Hayya, Lot streaming in multistage production systems. *International Journal of Production Economics*, 2000. **66**(3): 199-211.
28. Kalir, A.A. and S.C. Sarin, A near-optimal heuristic for the sequencing problem in multiple-batch flow-shops with small equal sublots. *Omega*, 2001. **29**(6): 577-584.
29. Bogaschewsky, R.W., U.D. Buscher, and G. Lindner, Optimizing multi-stage production with constant lot size and varying number of unequal sized batches. *Omega*, 2001. **29**(2): 183-191.
30. Kalir, A.A. and S.C. Sarin, Constructing near optimal schedules for the flow-shop lot streaming problem with sublot-attached setups. *Journal of Combinatorial Optimization*, 2003. **7**(1): 23-44.
31. Chen, J. and G. Steiner, On discrete lot streaming in no-wait flow shops. *Iie Transactions*, 2003. **35**(2): 91-101.
32. Van Nieuwenhuyse, I. and N. Vandaele. Minimizing total costs in flow shops with overlapping operations: the importance of gap costs. in Proceedings of the Annual POMS Conference POM in the Service Economy. 2003.

33. Chiu, H.-N., J.-H. Chang, and C.-H. Lee, Lot streaming models with a limited number of capacitated transporters in multistage batch production systems. *Computers & Operations Research*, 2004. **31**(12): 2003-2020.
34. TRUSCOTT, W.G., Scheduling production activities in multi-stage batch manufacturing systems. *International Journal of Production Research*, 1985. **23**(2): 315-328.
35. Vickson, R. and B. Alfredsson, Two-and three-machine flow shop scheduling problems with equal sized transfer batches. *The International Journal of Production Research*, 1992. **30**(7): 1551-1574.
36. Çetinkaya, F. C. and Kayaligil, M.S., Unit sized transfer batch scheduling with setup times. *Computers & industrial engineering*, 1992. **22**(2): 177-183.
37. Çetinkaya, F.C., Lot streaming in a two-stage flow shop with set-up, processing and removal times separated. *Journal of the Operational Research Society*, 1994. **45**(12): 1445-1455.
38. Laha, D. and S.C. Sarin, A heuristic to minimize total flow time in permutation flow shop. *Omega*, 2009. **37**(3): 734-739.
39. Glass, C.A. and E. Possani, Lot streaming multiple jobs in a flow shop. *International Journal of Production Research*, 2011. **49**(9): 2669-2681.
40. Wagneur, E. Lot streaming in no-wait flowshops with one machine never idle. in Control Applications, 2001.(CCA'01). Proceedings of the 2001 IEEE International Conference on. 2001. IEEE.
41. Chen, J. and G. Steiner, Lot streaming with detached setups in three-machine flow shops. *European Journal of Operational Research*, 1997. **96**(3): 591-611.
42. Bukchin, J., M. Tzur, and M. Jaffe, Lot splitting to minimize average flow-time in a two-machine flow-shop. *IIE Transactions*, 2002. **34**(11): 953-970.
43. Hall, N.G., G. Laporte, E. Selvarajah, and C. Sriskandarajah, Scheduling and lot streaming in flowshops with no-wait in process. *Journal of Scheduling*, 2003. **6**(4): 339-354.
44. Yoon, S.-H. and J.A. Ventura, Minimizing the mean weighted absolute deviation from due dates in lot-streaming flow shop scheduling. *Computers & Operations Research*, 2002. **29**(10): 1301-1315.
45. Pan, Q.-K. and R. Ruiz, An estimation of distribution algorithm for lot-streaming flow shop problems with setup times. *Omega*, 2012. **40**(2): 166-180.
46. Yoon, S.-H. and J.A. Ventura, An application of genetic algorithms to lot-streaming flow shop scheduling. *IIE Transactions*, 2002. **34**(9): 779-787.
47. Huq, F., K. Cutright, and C. Martin, Employee scheduling and makespan minimization in a flow shop with multi-processor work stations: a case study. *Omega*, 2004. **32**(2): 121-129.
48. Marimuthu, S., S. Ponnambalam, and N. Jawahar, Tabu search and simulated annealing algorithms for scheduling in flow shops with lot streaming.

Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2007. **221**(2): 317-331.

49. Marimuthu, S., S. Ponnambalam, and N. Jawahar, Evolutionary algorithms for scheduling m-machine flow shop with lot streaming. *Robotics and Computer-Integrated Manufacturing*, 2008. **24**(1): 125-139.
50. Marimuthu, S., S. Ponnambalam, and N. Jawahar, Threshold accepting and ant-colony optimization algorithms for scheduling m-machine flow shops with lot streaming. *Journal of materials processing technology*, 2009. **209**(2): 1026-1041.
51. Zhang, W., C. Yin, J. Liu, and R.J. Linn, Multi-job lot streaming to minimize the mean completion time in m-1 hybrid flowshops. *International Journal of Production Economics*, 2005. **96**(2): 189-200.
52. Martin, C.H., A hybrid genetic algorithm/mathematical programming approach to the multi-family flowshop scheduling problem with lot streaming. *Omega*, 2009. **37**(1): 126-137.
53. Buscher, U. and L. Shen. An integer programming formulation for the lot streaming problem in a job shop environment with setups. in Proceedings of the International MultiConference of Engineers and Computer Scientists. 2011. Citeseer.
54. Defersha, F.M. and M. Chen, A genetic algorithm for one-job m-machine flowshop lot streaming with variable sublots. *International Journal of Operational Research*, 2011. **10**(4): 458-468.
55. Karimi-Nasab, M., S. Seyedhoseini, M. Modarres, and M. Heidari, Multi-period lot sizing and job shop scheduling with compressible process times for multilevel product structures. *International Journal of Production Research*, 2013. **51**(20): 6229-6246.
56. Mortezaei, N. and N. Zulkifli, Integration of lot sizing and flow shop scheduling with lot streaming. *Journal of Applied Mathematics*, 2013. **2013**.