PARALLEL FLOW SHOP SCHEDULING WITH COMMON WORKSTATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY
MUHAMMET KEMAL ÇAKICI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

DECEMBER 2019

Approval of the thesis:

PARALLEL FLOW SHOP SCHEDULING WITH COMMON WORKSTATIONS

Submitted by **MUHAMMET KEMAL ÇAKICI** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural & Applied Sciences** ____________________

Prof. Dr. Yasemin Serin
Head of Department, **Industrial Engineering** ____________________

Assoc. Prof. Dr. Sedef Meral
Supervisor, **Industrial Engineering, METU** ____________________

Prof. Dr. Ferda Can Çetinkaya
Co-Supervisor, **Industrial Engineering, Çankaya University** ____________________

Examining CommitteeMembers:

Assoc. Prof. Dr. Canan Şerbetçioğlu
Industrial Engineering, METU ____________________

Assoc. Prof. Dr. Sedef Meral
Industrial Engineering, METU ____________________

Prof. Dr. Ferda Can Çetinkaya
Industrial Engineering, Çankaya University ____________________

Prof. Dr. Meral Azizoğlu
Industrial Engineering, METU ____________________

Assist. Prof. Dr. Sakine Batun
Industrial Engineering, METU ____________________

Date: 23.12.2019

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, Last name: Muhammet Kemal Çakıcı

Signature :

ABSTRACT

PARALLEL FLOW SHOP SCHEDULING WITH COMMON WORKSTATIONS

Çakıcı, Muhammet Kemal
Master of Science, Industrial Engineering
Supervisor: Assoc. Prof. Dr. Sedef Meral
Co-Supervisor: Prof. Dr. Ferda Can Çetinkaya

December 2019, 122 pages

In this thesis, we address the parallel flow shop scheduling problem with common workstations to minimize the makespan, considering the flow shop design defined solely by the number of operators at the workstations. We are motivated by the production environment of a compressor manufacturing company located in Konya, Turkey. Different from the similar studies in the literature, we use common workstations at some of the stages, prefer to place limited buffer areas between the stages, and do not allow for job crossing between the flow shops.

We develop two different Mixed Integer Linear Programming (MILP) models for this scheduling problem. These MILPs barely provide the optimal solution in almost one-hour time which is the time restriction in this study; hence, we propose several heuristic approaches that are based on the well-known NEH Algorithm. Moreover, two dispatching rules, SPT and LPT, are utilized to evaluate the results of the MILPs and the proposed heuristics better.

We perform extensive computational experiments using several problem instances that are differentiated by the number of operators and jobs, and compare the solution approaches. The results indicate that the proposed heuristic approaches are superior to the dispatching rules providing very close results to the MILP results in a short time and even better results as the number of jobs increases. In the experiments carried out for the compressor company's case, the proposed heuristic methods provide promising solutions that make it possible for the decision maker to select the most productive shop design doubling the production volume.

Keywords: Parallel Flow Shop Scheduling, Common Workstations, Makespan, Heuristic Approaches, NEH Algorithm

ÖZ

ORTAK İŞ İSTASYONLARIYLA PARALEL AKIŞ TİPİ ATÖLYELERİ ÇİZELGELEME

Çakıcı, Muhammet Kemal
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Sedef Meral
Yardımcı Tez Yöneticisi: Prof. Dr. Ferda Can Çetinkaya

Aralık 2019, 122 sayfa

Bu tezde, tüm işlerin tamamlanma zamanını en aza indirgeme hedefiyle paralel akış tipi atölye çizelgeleme problemine odaklandık ve bunu yaparken operatör sayısına bağlı olarak akış tip atölye tasarımlarını da inceledik. Literatürdeki çalışmalardan farklı olarak, belli aşamalarda ortak iş istasyonları kullandık, aşamalar arasında sınırlı tampon stok alanları yerleştirdik ve paralel atölyeler arasında iş geçişine izin vermedik. Konya'daki bir kompresör firmasının üretim ortamı bu çalışmayı yapma konusunda bizi motive etti.

Karışık tamsayılı doğrusal programlama (KTDP) olarak iki adet matematiksel model geliştirdik. Bu çalışmada çözüm süresi kısıtı olarak kullandığımız bir saat içerisinde KTDP nadiren en uygun çözümü bulduğu için, iyi bilinen NEH algoritmasının değiştirilmiş hali olan sezgisel yaklaşımlar önerdik. Ayrıca, KTDP ve sezgisel yöntemlerin sonuçlarını daha iyi değerlendirmek için iki tane öncelik kuralı sunduk.

Operatör ve iş sayılarına bağlı olarak farklı büyüklükteki problemlerle, çözüm yaklaşımlarını kıyasladığımız kapsamlı deneyler yaptık. Çıkan sonuçlara göre önerilen sezgisel yöntemler, öncelik kurallarından çok daha iyi sonuç verdi. Ayrıca sezgisel yöntemler çok kısa süre içerisinde KTDP modeline de çok yakın sonuçlar verdi ve hatta iş sayısının artmasına bağlı olarak daha iyi sonuçlar verdi. Firma için yapılan deneylerde ise önerdiğimiz sezgisel yöntemler üretim miktarını ikiye katlamayı mümkün kılan yüksek kaliteli çözümler sundu.

Anahtar Sözcükler: Paralel Akış Tipi Atölyeleri Çizelgeleme, Ortak İş İstasyonu, İşlerin Tamamlanma Zamanı, Sezgisel Yaklaşımlar, NEH Algoritması

To My Family and Friends

ACKNOWLEDGMENTS

First of all, I wish to express my deepest gratitude to my supervisor, Assoc. Prof. Dr. Sedef Meral for her guidance and encouragement throughout this research.

Besides my supervisor, I would like to thank my co-supervisor, Prof. Dr. Ferda Can Çetinkaya for his contributions and guidance during the thesis progress.

I also appreciate the support of my company, ÖZEN Compressor A.Ş, and thank my colleagues, managers, and especially İbrahim ÖZEN, CEO of ÖZEN Compressor A.Ş., for their great support, motivation, patience, and understanding throughout this study.

I want to thank my family and friends for their support, motivation, and faith during my Master of Science program.

Finally, I would also like to thank my wife and daughter for their great support, endless encouragement, and patience.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

MILP	Mixed Integer Linear Programming
PFS	Permutation Flow Shop Scheduling
NPFS	Non-Permutation Flow Shop Scheduling
SPT	Shortest Processing Time
LPT	Longest Processing Time
FIFO	First In First Out
LIFO	Last In First Out
FFS Based	Flexible Flow Shop Based
PFS Based	Parallel Flow Shop Based

CHAPTER 1

INTRODUCTION

The companies need to change and improve the way they do their businesses following the environmental conditions and requirements. In the past, products were at the focus of company owners and managers, and all employees made their efforts to produce better products. However, over time, the pure production has ceased to be the primary concern as before, and the customer has gained more value associated with the production competency that most of the companies have acquired. Thus, in today's highly competitive environment, manufacturing companies have to be more customer-oriented to come into prominence. While, in the past, producing quality or cheap products used to be more than sufficient to make a difference, now the customers' customized products need to be manufactured in short lead time, at a low price and the required quality.

It is not easy for the companies to ensure customer satisfaction considering this wide range of products and limited resources of the companies such as machines, operators, and raw materials. To increase and maintain customers' satisfaction, companies need to build advanced production systems which allow them to establish more flexible structures, to become more appropriate for customization of customers' orders, and thus to respond more easily and quickly to customers' highly varying demands.

Within the scope of the advanced production system, optimal or very-close-to-optimal production plans are required to shorten the customer lead time. Several flow shop design and operational issues need to be addressed taking into account practical restrictions to end up with production scheduling: how the flow shops are to be designed and laid out, how many operators work at the workstations, in which type of flow shops and sequences the products are to be produced. These issues can be dealt with depending on the shop design configuration selected as either a flow shop, or a parallel flow shop or a flexible flow shop.

Flow shop refers to a production process where all products follow the same linear route through the shop. Each product enters the shop at the first stage and leaves the shop from the last stage once its operation is finished. Assume that there are *n* jobs (products) each requiring a set of operations to be completed on *m* machines (stages) lined up in series. Each job needs to pursue the same routing through *m* stages, and to be processed exactly once at each stage. The processing times of *n* jobs can be different for a specific operation that is to be performed at the same stage. When an operation of a job is started at a stage, it must be finished without any interruption. Besides, a buffer area (queue space) is placed between any pair of adjacent stages, which enables storing of WIP products between stages at the shop; the buffer area can either be infinite or finite depending on the size of product and also the available area at the manufacturing hall. In the case of a totally full buffer area, the upstream machine is blocked and therefore cannot release the job into the buffer after completing its processing.

At times, in an attempt to increase the production capacity, companies select to install additional independent flow shops, which are almost the same as the existing flow shops. In the literature, this shop is called as “parallel flow shop”, which is very similar to a flow shop and can be considered as the parallel placement of more than one independent flow shop. All the assumptions about the flow shop also apply to the parallel flow shop.

Flow shop and parallel flow shop are very suitable production systems for companies that produce similar products requiring the same set of operations performed with the same routing. However, over time, in line with the proliferating product variety and product demand, companies have started to increase their throughput and flexibility by constructing more sophisticated flow shops that have parallel machines at some or all of the stages. Thus, another flow shop design or configuration has emerged which is called as “flexible flow shop” or also referred to as “hybrid flow shop”.

A flexible flow shop consists of several processing stages in series with infinite or finite buffer areas between adjacent stages like in flow shops. However, there are several machines in parallel at one of the stages at least, and products must be processed by at most one machine at each stage. In the same way as in the flow shop, each product visits the stages in the same order, and once its processing is completed on a machine at a stage, it is transferred directly to either an available machine at the next stage or an available buffer ahead of the current stage unless any of the next stage's machines is available. If the WIP product cannot be transferred either to the next stage or to the buffer ahead, it remains at the current stage blocking the machine until the buffer becomes available. Thus, this blocking prevents the processing of other products on the blocked machine.

Depending on the constraints and needs of the company, such as the number of employees, available space, available buffer area, machine or tool restrictions; different types of flexible flow shops can be designed and operated. Given the design of the flexible flow shop, the main approach to its operation consists of two problems to be solved in succession: first, allocating operations of the jobs to the parallel machines or a single machine at the stages; second, sequencing the jobs on the machines. Thus the shop schedule can be obtained with the solution of these two problems.

In this study, a new approach to the flow shop scheduling problem is proposed so as to increase the throughput rate of the shop over time which is equivalent to minimizing the makespan for a set of jobs. Due to the physical characteristics of both the manufacturing environment and the jobs in the practice company that we are motivated by, several flow shop designs are analyzed that turn out to be a hybrid of parallel flow shops and flexible flow shops: a parallel flow shop with common workstations. Hence, we consider two parallel flow shops which have common workstations at some of the stages. While the workstation at any stage of the flow shop normally operates only the products that are assigned to that flow shop, the common workstation of the flow shops processes the products from both of the flow shops. However, despite the common usage of some workstations by both flow shops, a product cannot cross over to the other flow shop from its original flow shop due to its size and voluminous body and therefore has to leave from its original flow shop that it was assigned to at the first stage. For the parallel flow shop with common workstations as defined above, we formulate the scheduling problem for a given mix of jobs (customer orders) as a Mixed Integer Linear Programming model. Due to the computational complexity of the model, we develop several heuristic approaches based on the NEH algorithm to obtain satisfactory -if not optimal- solutions in relatively shorter times. We try different parallel flow shop designs given the number of employees that can work at the workstations and obtain the best makespan schedule for them. The parallel flow shop design with common workstations that completes the given set of jobs at the minimum completion time is selected as the best shop design with the numbers of stages and their workstations, and common workstations précised.

The chapters of the thesis are organized as follows.

In Chapter 2, a brief explanation about the flow shop, parallel flow shop and flexible flow shop scheduling problems is given. Then, the characteristics of our problem are defined with all its aspects such as the manufacturing environment, tasks of jobs, workstation attributes, and some operational restrictions as well. Besides, the assumptions that we consider in formulating the problem are stated.

In Chapter 3, the literature survey is presented. Previous studies that have inspired us with their different approaches are analyzed. The studies are categorized according to some features. Researches on scheduling problems in flow shops, parallel flow shops and flexible flow shops are examined separately. Moreover, the studies are detailed separately in terms of their exact solution and heuristic approaches.

In Chapter 4, solution approaches are explained in detail. First, we present two MILP (Mixed Integer Linear Programming) models for our problem by extending the mathematical models posed by Sawik (2000) and Ribas (2017) for flexible and parallel flow shops, respectively.

On the other hand, since the scheduling problem we address is NP-hard for all traditional optimality criteria even when setup times are negligible, the computational time of optimal solution procedures is prohibitively long and exceeds the acceptable limit when medium or large-sized instances are considered (Garey et al., 1976; Kis and Pesch, 2005). Therefore, despite the best solution that it provides, the mathematical model is mostly a hard way to implement in real-life cases. Hence, in the second part of this chapter, two heuristic approaches, specifically modified NEH algorithms, are proposed as alternative approaches. These two methods rapidly reach a solution close to the solution provided by the mathematical model, making them appropriate and practical methods, especially for real-life problems. Furthermore, in the third part of this chapter, we propose the use of two well-known dispatching rules, SPT (Shortest Processing Time) and LPT (Longest Processing Time). Thus, we ensure to compare the proposed solution methods with the commonly used dispatching methods and evaluate them accordingly.

In Chapter 5, a computational study is performed with numerous experiments considering 15, 25, and 50 jobs. First, two MILP models are compared, and then the performances of the heuristic approaches are analyzed against the solutions provided by the MILP model that gives the better results. The effectiveness of each heuristic approach is evaluated regarding the closeness of its result to that of the MILP model (solution quality) and the computational time.

In Chapter 6, a case study is represented from the ÖZEN Compressor Company in the machinery industry in Konya. ÖZEN Compressor is engaged in manufacturing air compressors as one of the leading companies in the sector in Turkey. The wide range of products includes screw compressors, piston compressors, air tanks, dryers, and filters. The company desires to increase their production volume, specifically for the screw compressors, and this has inspired us in the thesis study. Therefore, in this chapter, we compare our solution approaches' results with the real situation at the company in order to observe the improvements in terms of makespan.

Finally, we conclude in Chapter 7 and also present some suggestions for the future research issues.

CHAPTER 2

PROBLEM DEFINITION

In this section, first, flow shop scheduling, flexible flow shop scheduling, and flexible flow shop scheduling problems and their environments are examined in detail. Afterward, our problem is defined by all its environmental characteristics and assumptions.

2.1. Flow Shop Scheduling

A primary flow shop consists of several stages in series with only one machine at each stage. Jobs enter the system from the initial machine and then pursue the machines in order and exit the system from the last machine. All jobs have to follow the same processing route (Figure 2.1).

Figure 2.1 A Typical Illustration of a Flow Shop

Each job consists of several separate tasks called operations. Therefore, to complete a job, all of its operations must be executed in a certain order as defined by the precedence relationships among the operations of the job. In the simplest flow shop, there are n jobs and m machines, and each job consists of exactly m operations, and the i^{th} operation of each job must be carried out on the i^{th} machine where no one machine can perform more than one operation simultaneously. The processing time of job j on machine i is t_{ij}, defined for each operation of each job. However, in the more complex flow shops, some jobs may skip some of the operations in the order without changing the common routing.

Typically, there is no buffer area between the machines (workstations) in flow shops. The job that is completed at a workstation passes directly to the next one. However, if the next workstation is occupied by another job, then the job cannot move to it and remains at the workstation where its operation is completed until the next workstation becomes available. During this period, the workstation is blocked by the job, and hence the next job on the sequence cannot be started at the blocked workstation although it may be available for processing. However, in more improved flow shops, a buffer area of any required size can be placed between every two adjacent workstations allowing the WIP jobs to be stocked up therein when the next workstation is occupied. The capacity of the buffer area depends on the product (job) sizes and available area at the flow shop, as well as the throughput rate required from the flow shop. If the capacity of the buffer area is limited and turns out to be full at times, then the blockage situation affects the throughput rate adversely. Otherwise, if the capacity of the buffer area is infinite, which means all jobs can be stored at the same time therein, no blockage occurs at any workstation.

Furthermore, in the flow shops, if all jobs pass through the stages in the same sequence, then the schedule is called a permutation flow shop schedule (PFS). However, in cases where the job sequence changes from one stage to another one, the schedule is called a non-permutation flow shop schedule (NPFS). Figure 2.2 illustrates an example for both PFS and NPFS with two jobs and four stages (machines). NPFS scheduling has a larger solution space than PFS scheduling, since PFS scheduling searches its optimal solution among $n!$ feasible schedules while NPFS scheduling has to consider $n!^m$ feasible sequences. In other words, the solution space of PFS scheduling problem is a subset of the solution space of NPFS scheduling problem (Rossit et al., 2018). However, it should be kept in mind that the capacity of buffer area between workstations should be at least as many as the number of jobs, which may correspond to an infinite space when NPFS is preferred.

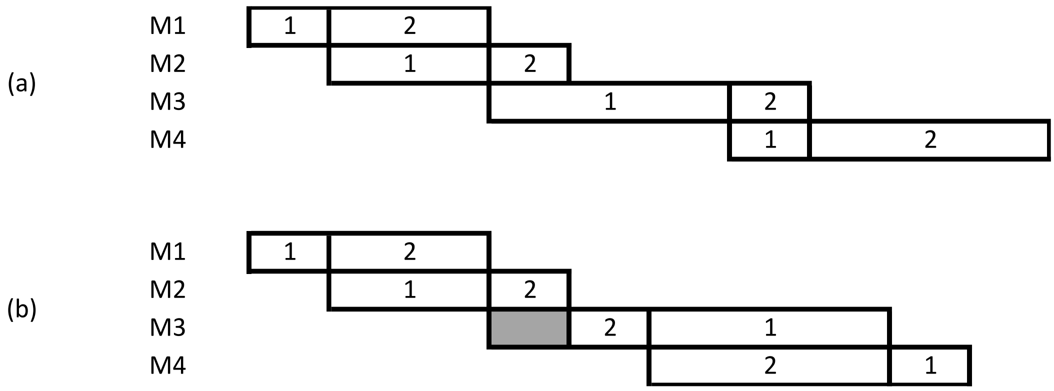

Figure 2.2 (a) Permutation Flow Shop (PFS); (b) Non-permutation Flow Shop (NPFS)

Different objectives can be defined in flow shop scheduling. In literature, various objective functions are used, such as minimizing the number of workstations, minimizing the makespan, maximizing the line efficiency, minimizing the total tardiness, and minimizing the average sum of earliness and tardiness. Even multiple objectives can be used, which allows us to consider more than one objective at a time for a flow shop scheduling problem. However, minimizing makespan (C_{max}) is a commonly used objective function, in about 60% of the studies in the literature (Ruiz and Vázquez-Rodríguez, 2010). Through makespan minimization, we can shorten the total time needed to complete all jobs and thus increase the efficiency of the flow shop.

2.2. Parallel Flow Shop Scheduling

The parallel flow shop is composed of more than one parallel flow shop and holds all the characteristics of flow shop scheduling (Figure 2.3). When the capacity of the flow shop becomes insufficient, adding the same or similar flow shop(s) can be preferred as a solution, resulting in a parallel flow shop configuration.

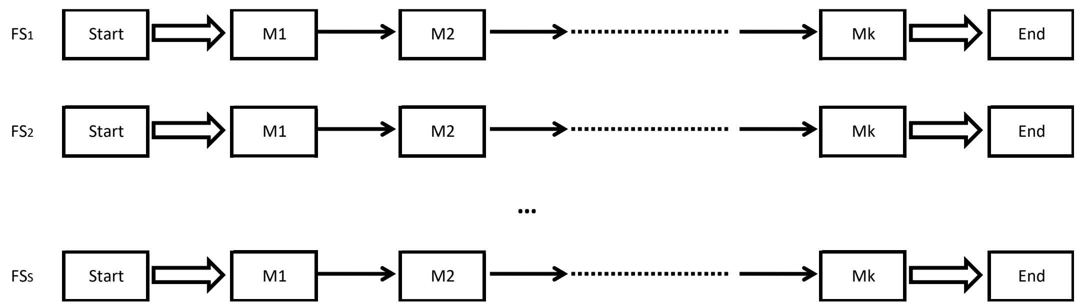

Figure 2.3 A Typical Illustration of a Parallel Flow Shop

Different from the flow shops, first, all jobs need to be assigned to one of the flow shops in the parallel flow shop scheduling problem. Then, the jobs in each flow shop are scheduled in order to optimize the objective function.

On the other hand, the flow shops arranged in parallel can be exactly the same or slightly different from each other. For instance, the number of the stages at each flow shop or the capacity of the buffer area between two consecutive machines at each flow shop can be the same or different. Besides, some special equipment or machines can be used at a particular flow shop, and for this reason, at least some of the jobs may need to be assigned to those specific flow shops. Nevertheless, after the assignment of jobs to the flow shops, the overall functioning of the parallel flow shop is exactly the same as that of the flow shop, since each flow shop can be scheduled independently from the others and operates independently.

2.3. Flexible Flow Shop Scheduling

The flexible flow shop is a more advanced form of the classical flow shop. Similar to flow shops, flexible flow shops consist of several stages in series where jobs start to be processed at the first stage, follow the next stages, and finish its processing at the last stage. Each job consists of several operations that have to be operated at a different stage, and the order of operations is the same for all jobs. Likewise, an infinite or limited buffer area can be placed between two adjacent stages, and the same blocking case is experienced if the next stage is occupied or the buffer area ahead is full. Even though the sequence of jobs can be different or the same from one stage to the other depending on the scheduling problem, the routings of the jobs are the same.

On the other hand, there are significant differences between flexible flow shops and classical flow shops. While there is only a single workstation at each stage in the flow shop, there may be more than one parallel workstation at each stage in flexible flow shops (Figure 2.4). When a job reaches any stage, its operation has to be processed by one of the parallel workstations therein. For instance, if a flexible flow shop has *n* jobs with *m* operations and *m* stages with *k* workstations at each stage, then the i^{th} operation of job *j* is performed at one of the available *k* workstations at the i^{th} stage, and the processing time of job *j* for that stage *i*, t_{ij}, is the same for all *k* workstations at stage *i*. As in flow shops, some jobs may skip some stages in flexible flow shops and processing times for the skipped stages are taken as zero.

In the scope of flexible flow shop scheduling, allocating more workstations to some stages, especially the bottleneck ones, compared to the other stages is possible, and in this way, the completion time of all jobs can be shortened, which is the most important advantage of a flexible flow shop compared to a flow shop.

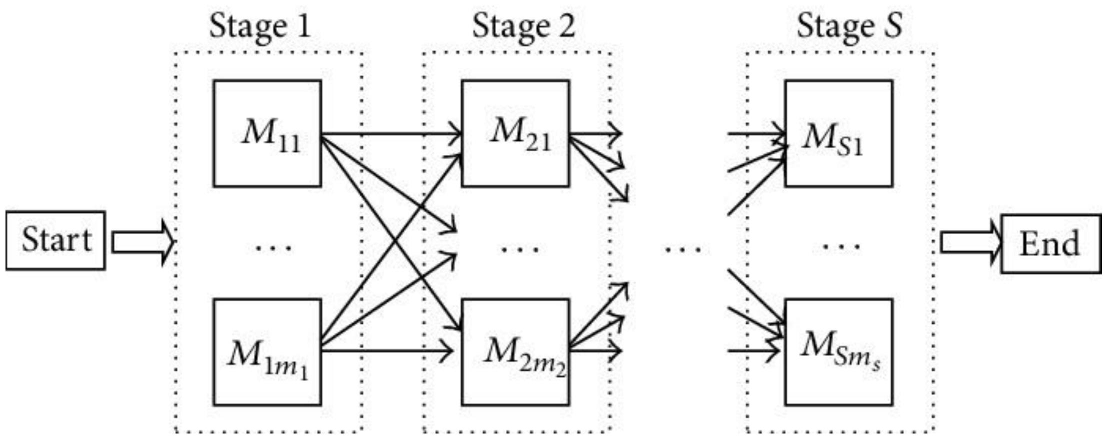

Figure 2.4 A Typical Illustration of a Flexible Flow Shop

2.4. Problem Environment

In this thesis, we focus on the scheduling problem at a parallel flow shop with some common workstations at least at one stage of the shop. In this parallel flow shop, a set of jobs are processed, each job corresponding to a certain model of a product (models of the product “screw compressor” in the company case). This means that more than one product model can be processed at the same flow shop, rather than the same product being repetitively processed as in single-model flow shops.

In this study, we prefer parallel flow shops fixing the shop with two parallel flow shops only, considering the operational restrictions of the practice company. The parallel flow shop that we plan to design looks similar to the flexible flow shops as shown in Figure 2.5, but with one major difference: jobs crossing over between flow shops is not allowed in the manufacturing environment we consider, while it is allowed at flexible flow shops. This means that when a product enters one of the flow shops at the beginning, it cannot pass to the other flow shop later and should leave from the same flow shop at the last stage.

Total work content of jobs may vary from one job to another, and likewise, processing time of the same operation may be different from one job to another. In that case, constituting parallel workstations at the bottleneck stage(s) can be a good solution to somehow balance the stages at the shop. Similarly, a workstation may be processing an operation on all jobs with a relatively smaller processing time compared to the individual processing times at the other workstations. Hence, only a single workstation at a certain stage of the parallel flow shop may be sufficient for processing the operation assigned to it for all jobs flowing through the parallel flow shops (see Figure 2.5 with two parallel flow shops and three common workstations, processing two different jobs -red and green- only). When there is more than one available job at the common workstation, they are processed according to the FIFO rule. The use of a common workstation allows us to use fewer operators and hence to reduce idle time. Using common workstations is the key characteristic of the parallel flow shop considered in this study, differentiating it from the traditional parallel flow shops.

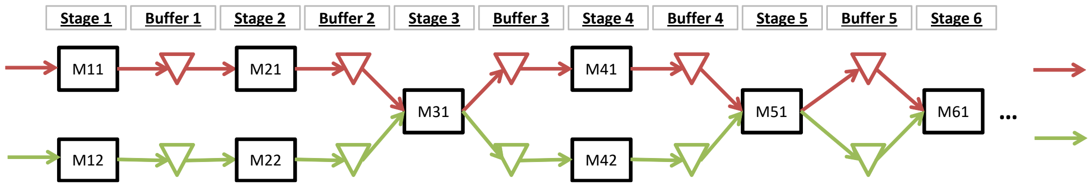

Figure 2.5 A Typical Illustration of Parallel Flow Shops with Common Workstations

Processing times of a job at the stages are different which is valid for all jobs in general. Hence blocking and starvation may occur at the workstations unless buffers are allowed between the stages. For instance, while an operation of a job is performed at the second stage in 20 minutes, at the same time, the first operation of another job is performed at the first stage in 15 minutes. This means that the job at the first stage needs to wait for a minimum of 5 minutes to reach the second stage, which is occupied with the former job in the sequence. So, the buffer area between two consecutive workstations at the stages in the flow shops provides storage for the WIP jobs and thus prevents the blockage at the former workstation by the latter. The capacity of the buffer area can be determined based on the available area. Unlike the real-life cases, the buffer area is generally considered infinite in the literature (Rossit et al., 2018; Ruiz and Vázquez-Rodríguez, 2010). Since the infinite buffer area is not feasible in real-life cases, we prefer to limit the buffer area between any pair of workstations with one WIP product (job) in our company case. In this case, if the buffer area is empty when an operation of a job is completed at a workstation, the job can leave the workstation and sit at the buffer, unless the next workstation is available. Otherwise, a blockage is inevitable at the former workstation caused by the latter.

Furthermore, most of the studies in the literature consider permutation flow shop scheduling, which does not allow for sequence changes from one workstation to another, while the rest analyses non-permutation flow shop scheduling. For some practical reasons like the size of the product and others discussed below, we consider the permutation flow shop scheduling in this study. First, we have limited buffer areas with one-job area only between workstations, which does not allow for implementing non-permutation flow shop schedules. Second, permutation flow shop scheduling is more common and easier to control in real life. By implementing PFS scheduling, we ensure the same job sequence at all workstations at each flow shop except the common workstations, where we use the FIFO rule and the job sequence therein turns out to be a mix of the jobs coming from the two parallel flow shops.

In the shop considered, production is operator-paced, and hence processing times may vary with operators' performance that depends on their skill level, motivation, and experience. Hence, in such operator-paced flow shops, several measures for processing times are used, such as ranges to indicate minimum and maximum values or mean value and standard deviation of processing times. In this study, we assume that we have multi-skilled and sufficiently experienced operators who can perform the operations on the jobs with only a negligible deviation from the mean processing time. Therefore, although it may not be realistic in some real-life cases, we assume constant processing times for all operations of jobs.

Besides, some operations require particular types of equipment or tools such as crane or welding robots. Therefore, these operations cannot be assigned to every workstation, but can only be assigned to the workstations that include the required equipment or tools. However, in our study, we ignore this case and assume that no special equipment or tool is necessary for any of the operations. So, we have no constraint but only the precedence constraints among the operations of a job while assigning operations of jobs to the workstations. Precedence constraint means that there are several operations to be performed on a job with a certain precedence that should be obeyed in the assignment of operations to the workstations. In our company case the precedence relationship is the simplest one which is a linear relationship starting with operation 1 at stage 1 and ending with operation m at stage m in series.

Depending on the total number of operators, the flow shop design is basically defined by the number of workstations arranged in two parallel flow shops. Since we have two parallel flow shops in this study, there can be one or two operators at each stage. However, for the number of stages, we do not have any restrictions, except the number of operators. For instance, in the case of 4 operators at the shop, we have the following five alternative flow shop designs:

i. 2-2: That means two operators at the first stage and two operators at the second stage. In this case, there is not any common workstation, and hence the shop consists of two parallel flow shops.
ii. 2-1-1: That means two operators at the first stage, one operator at the second stage, and one operator at the third stage. In this case, the second and third workstations are the common workstations.
iii. 1-2-1: That means one operator at the first stage, two operators at the second stage, and one operator at the third stage. In this case, the first and third workstations are the common workstations.
iv. 1-1-2: That means one operator at the first stage, one operator at the second stage, and two operators at the third stage. In this case, the first and second workstations are the common workstations.
v. 1-1-1-1: That means one operator at each stage. In this case, the shop turns out to be a flow shop with four workstations in series.

In this study, while looking for the best shop design providing the minimum makespan for a set of jobs, given the number of operators available, we consider all possible shop designs that are only restricted by the given number of operators.

Finally, in this study, we prefer to use "minimizing makespan" as an objective function, thus, we can minimize the total time needed to complete all jobs that are the orders received on a weekly basis.

In addition to the issues discussed above, there are some other practical/operational issues that need to be taken into account as assumptions:

- Orders for products are accumulated on a weekly basis, and constitute the jobs to be processed in the following week. Thus the jobs have a common due date that is the end of the week.
- Orders for products cannot be cancelled after being scheduled, and any new order is not acceptable to be inserted into the already completed schedule of

the current week. This practice means that after the weekly schedule is completed, it is frozen and no update is allowed.

- Each product can be produced at any of the parallel flow shops.
- The precedence diagram of operations and processing times of them are known and constant. But some jobs may skip some of the stages, thus some operation times may be zero for some jobs.
- Operators are multi-skilled (flexible operators), and it does not matter who works at which workstation.
- Each operation of each job can be assigned to a workstation in line with the precedence diagram.
- Each operation must be performed at only one workstation (an operation cannot be split between workstations).
- Only one operation can be processed at a time at a workstation.
- Returning back and revisiting the previous stages is not allowed.
- Transportation times between stages are negligible.

In brief, we study the company case, where only two parallel flow shops with common workstations and limited buffer areas between stages are to be established, and aim at providing the optimal or near-optimal job schedule in terms of makespan. In the search for the best schedule, PFS schedules are considered and jobs crossing over between the flow shops is not allowed due to the product characteristics.

CHAPTER 3

LITERATURE REVIEW

3.1. Overview

Flow shop scheduling, parallel flow shop scheduling and flexible flow shop scheduling problems have been widely discussed in the literature and continue to be studied. Most of them approach the problems with different aspects, as constraints and objectives. Thus, each study can bring a solution for various flow shop scheduling problems in the industry.

In these studies, generally, either the mathematical model that has already been developed is improved or a different heuristic method is proposed. Thus, the better results are progressively obtained, and these studies inspire other studies and improvements.

For this purpose, studies are carried out for numerous objectives. As shown in Figure 3.1 below, the objective function of the majority of the hybrid (flexible) flow shop scheduling problems refers to makespan minimization. The other commonly used objective functions are total/average completion or flow time, total/average tardiness, and number of tardy jobs.

On the other hand, various solution methods have been used in studies with different objective functions. Although numerous studies prefer exact solution methods such as integer programming formulation or B&B, heuristic or metaheuristic approaches have more widespread use in the studies. The most important reason for this situation is the fact that the flow shop scheduling problem or flexible flow shop scheduling problem is an NP-hard problem requiring prohibitively long computational times. For this reason, heuristic methods are more preferred in the hybrid flow shop scheduling studies to obtain optimal or near-optimal solutions within an acceptable computational time (Figure 3.2).

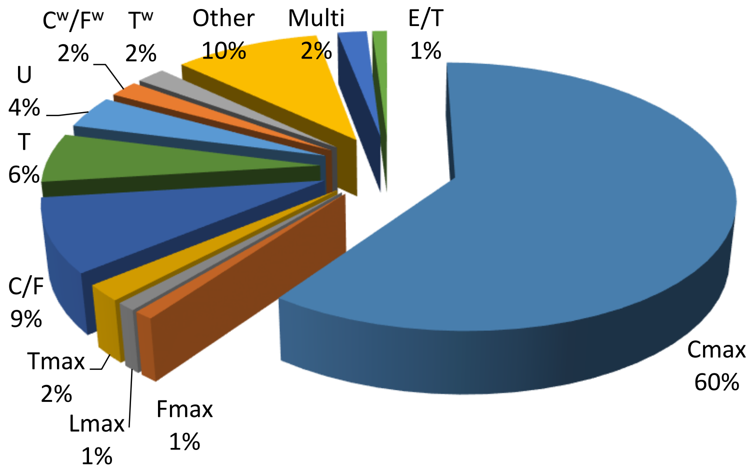

Figure 3.1 Distribution of Objective Functions in Flexible Flow Shops

(C_{max} : Makespan, F_{max} : Maximum flow time, L_{max} : Maximum lateness, T_{max} : Maximum tardiness, C/F : Total/average completion or flow time, T : Total/average tardiness, U : Number of tardy jobs, C^w/F^w : Total/average weighted completion or flow time, T^w : Total/average weighted tardiness, Multi : Multiples objectives, E/T : Total/average sum of earliness and tardiness(Ruiz and Vázquez-Rodríguez, 2010))

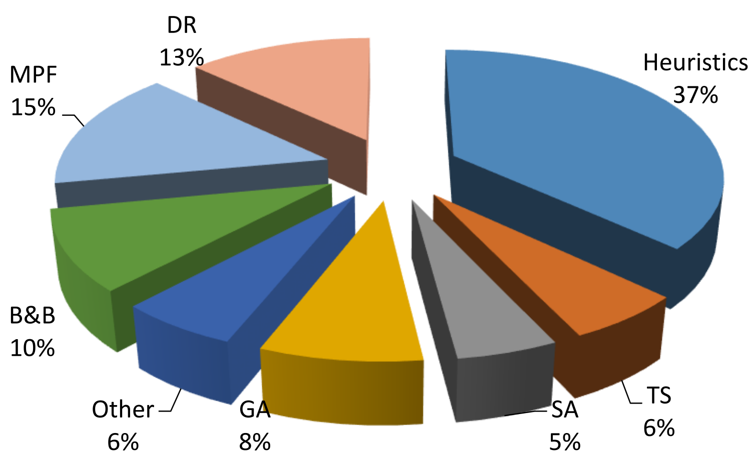

Figure 3.2 Distribution of Implemented Techniques in Flexible Flow Shops

(TS :Tabu search, SA : Simulated annealing, GA : Genetic Algorithms, B&B : Branch and Bound, MPF : Mathematical Programming and Formulation, DR : Dispatching rules (Ruiz and Vázquez-Rodríguez, 2010))

As mentioned before, many studies have been carried out about flow shop scheduling, parallel flow shop scheduling, and flexible flow shop scheduling problems. Nevertheless, the studies which inspire us most within the scope of the thesis are presented in this chapter.

3.2. Flow Shop Scheduling Problem

There are numerous studies about flow shop scheduling in literature. However, in this thesis, due to its similarity to our problem, we especially focus on flow shops with multiple machines.

Campbell et al. (1970) propose a heuristic algorithm to find out the minimum total processing time (makespan) for the *n* job, *m* machine flow shop sequencing problem where there is only one machine at each stage.

Nawaz et al. (1983) develop a heuristic algorithm, named NEH, to solve the *n* job *m* machine flow shop scheduling problem, and then compare it with fifteen other algorithms. Computational studies show that the proposed NEH algorithm performs especially well on large flow shop problems in both static and dynamic sequencing environments.

Gupta (1988) presents an efficient heuristic algorithm that uses Johnson's algorithm while sequencing jobs for the two-stage flow shop scheduling problem, where there are identical multiple machines at stage 1 and there is only one machine at stage 2, and the objective is the minimum makespan.

Taillard (1990) compares classical heuristics to solve the flow shop scheduling problem. The NEH algorithm is found to be superior to the other heuristic approaches. Then, the tabu search algorithm is applied to the problem, which allows us to get a better solution than NEH.

Chen (1995) addresses the two-stage flow shop scheduling problem with parallel machines at one stage only. Some efficient heuristic algorithms where worst-case performance ratio is used are proposed for finding a minimum makespan schedule.

Agnetis and Arbib (1997) propose polynomial-time dynamic programming algorithms for a flow line assembly problem to minimize makespan or inventory costs subject to various constraints such as limited buffer capacity, set-up time and transportation time. The problem concurrently addresses both assigning the operations to the machines and sequencing them.

Dessouky et al. (1998) present a branch and bound procedure and a heuristic procedure for flow shop scheduling problems which includes two stages and three stages with uniform parallel machines and has an objective function of makespan minimization.

Brah and Wheeler (1998) compare scheduling rules in a flow shop with multiprocessors. They examine the effects of problem characteristics (number of jobs, number of stages and number of parallel processors at each stage) and localized scheduling rules on mean flow time and makespan. The "shortest processing time" rule is found to be consistently superior, and its superiority is observed to be higher for the mean flow time than for the makespan.

Nowicki and Smutnicki (1998) implement a tabu search approach with a specific neighborhood definition which employs notions of critical path in a graph and a block of jobs to find a minimum makespan in a flow shop with parallel machines.

Brah and Loo (1999) aim to minimize makespan in a flow shop with multiple processors by applying five heuristic algorithms in the literature: NEH algorithm, Palmer’s algorithm, CDS1, CDS2, and Ho algorithm. Besides, the effects of problem characteristics such as the number of jobs, number of stages and number of parallel processors at each stage as well as the performance of scheduling heuristics are investigated by using regression analysis.

Schuurman and Woeginger (2000) investigate the two-stage multiprocessor flow shop scheduling problem, where the numbers $m1$ and $m2$ of machines available in the two stages are part of the input, and the objective function is makespan minimization. Polynomial-time approximation scheme is derived for the two-stage multiprocessor flow shop scheduling problem.

Wu et al. (2003) study flow shop scheduling problem with parallel machines at each stage and objective function of makespan minimization. A genetic algorithm is presented for solving the scheduling problem considering the special constraints that require some jobs to be processed on some certain machines.

Semanco and Modrak (2012) propose a constructive heuristic approach that employs Johnson's rule to give a good initial solution for the improvement heuristic, also known as metaheuristics for the solution of the n-job m-machine permutation flow shop scheduling problem. The proposed heuristic algorithm is tested against four other heuristics, namely, NEH, Palmer's Slope Index, CDS and Gupta's algorithm for minimizing makespan.

Hossain et al. (2014) aim to minimize makespan in flow shop scheduling. Palmer's heuristic, CDS heuristic, and NEH algorithm are presented for solving the flow shop scheduling problem to minimize makespan. NEH yields better quality results as compared to Palmer and CDS heuristic.

Hojati (2016) focuses on a 2-stage m-machine (one machine at stage 1, m machines at stage 2) disassembly flow shop scheduling problem for which three heuristic approaches are used to minimize makespan.

Laribi et al. (2016) deal with makespan minimization in the flow shop scheduling problem where no renewable resource constraints are considered. The importance of Johnson's algorithm for the two-machine flow shop under resource constraints is illustrated, and a mathematical model is also presented for the two-machine flow shop scheduling. Then a well-known heuristic algorithm, NEH Algorithm, is adapted to propose a fast solution for the m-machine flow shop problem subject to resource constraints.

Kazemi et al. (2017) investigate the two-stage assembly flow shop scheduling problem with a batched delivery system where there are m independent machines at the first stage doing the components of a job and multiple identical assembly machines at the second stage, each of which can assemble the components and complete the job. The objective is to schedule the jobs, to form them into batches to minimize the sum of tardiness plus delivery costs. A mixed-integer non-linear mathematical model is presented together with the imperialist competitive algorithm (ICA) and a hybrid algorithm (HICA) for this problem.

Bai et al. (2017) prefer to use a mixed-integer programming model and B&B algorithm in order to acquire the optimal solution while studying a permutation flow shop scheduling to minimize makespan.

Khorasanian and Moslehi (2017) investigate a 2-machine flow shop scheduling problem with blocking, multi-task flexibility of the first machine, and preemption constraints. In this study, two mathematical models and two heuristic algorithms are developed for makespan minimization.

Dong et al. (2017) consider the flow shop scheduling problem, where *m* parallel identical machines and two stages are located to process *n* jobs, and fully polynomial-time approximation scheme (FPTAS) is used in order to minimize makespan.

Tong et al. (2018) consider minimizing makespan for *m*-identical machine, *k*-stage flow shop scheduling with a set of jobs. Each job can be processed by any one of the flow shops, but switching between flow shops is not allowed. A polynomial-time approximation scheme is presented for this problem.

Rossit et al. (2018) investigate the studies about the flow shop scheduling problem, both permutation flow shop and non-permutation flow shop, and then classifies the studies according to their objective functions such as completion-time based objective, due-date based objective, experimental mono-objective, multi-objective, and economic objective. Furthermore, a list of recent studies is given considering the solution methods and constraints that are used in these studies.

3.3. Parallel Flow Shop Scheduling Problem

In the literature, there are fewer articles about parallel flow shop scheduling than the flow shop and flexible flow shop scheduling. The studies that inspire us are given below.

Sundararaghavan et al. (1997) investigate parallel flow shop scheduling problems with the objective of minimizing makespan. Once a job is assigned to a line, it has to be processed on that line and cannot cross to the other flow shops. Our problem looks like this problem, but unlike this study, we have common workstations that the flow shops can use. While an exact procedure for smaller problems is used, heuristic and simulation approaches are developed to solve large scale problems.

Al-Salem (2004) addresses the problem of minimizing makespan on a two-stage m proportional parallel flow shops with n independent jobs and proportional processing times. A multi-phase heuristic algorithm to minimize makespan on two proportional parallel flow shops is proposed, and a simulation study is conducted to examine the effectiveness of the proposed heuristic algorithm for small and large size problems.

Zhang and Velde (2012) consider the NP-hard problem of scheduling n jobs in m two-stage parallel flow shops so as to minimize makespan. This problem decomposes into two subproblems: assigning the jobs to parallel flow shops; and scheduling the jobs assigned to the same flow shop by the use of Johnson's rule. For the cases of m=2 and m=3, an approximation algorithm is proposed.

Ribas et al. (2017) consider the NP-hard problem of scheduling n jobs in F identical parallel flow shops, each consisting of a series of m machines. A mathematical model along with some constructive and improvement heuristics are proposed to solve the parallel blocking flow shop problem, and thus to minimize makespan among flow shops. On the other part, an iterated local search (ILS) and an iterated greedy algorithm (IGA), both of which are combined with a variable neighborhood search (VNS) are proposed as a heuristic approach. This paper is crucial for our study since the proposed mixed-integer programming formulation in this paper has inspired us to generate a mixed-integer linear programming formulation in the scope of our study.

3.4. Flexible Flow Shop Scheduling Problem

Flexible flow shop scheduling problem has been examined in the literature in a wide range with different aspects. The studies related to our problem are summarized as follows.

Wittrock (1985) focuses on a flexible flow shop problem with an objective function of makespan minimization. Along with an LP as a mathematical model, a dispatching rule called LPT and a dynamic balancing algorithm as a heuristic approach are presented to obtain the solution.

Kochhar and Morris (1987) focus on flexible flow shop scheduling problem with constraints such as setup times, finite buffers blocking and starvation. Dispatching rules and heuristic algorithms such as myopic and Local Search (LS) methods are applied in order to obtain the minimum average interval (the interval of a job, also referred to as its sojourn time, being the time it spends in the system, from entering the first workstation to leaving the last one).

Rajendran and Chaudhuri (1992) introduce a Branch and Bound algorithm to minimize the total flow time for permutation flexible flow shop scheduling with parallel processors at stages. A heuristic algorithm is presented for the two-stage multi-processor flexible flow shop. As a result of computational study, the proposed heuristic algorithm comes up with a near-optimal solution.

Gupta et al. (1997) consider a non-preemptive two-stage hybrid flow shop problem in which the first stage contains several identical machines, and the second stage contains a single machine. The objective is to find a schedule that minimizes makespan. Several lower bounds are derived and tested in a Branch and Bound algorithm. Also, constructive heuristics and descent algorithms are proposed.

Haouari and M’Hallah (1997) focus on the two-stage hybrid flow shop scheduling problem where parallel machines are placed at each stage. A simulated annealing based heuristic algorithm and a tabu search based heuristic algorithm are proposed while seeking the minimum makespan.

Grangeon et al. (1999) compare dispatching rules, namely FIFO, LIFO, LPT, and SPT to minimize makespan in hybrid flow shops that consist of k stages and m parallel machines at each stage.

Linn and Zhang (1999) has a survey paper and explains the studies about the hybrid flow shop scheduling problem by grouping them into three categories: Two-stage HFS, Three-stage HFS, and k-stage (k>3) HFS. Finally, certain suggestions are given for future research directions.

Sawik (2000) presents a mixed integer programming formulations for scheduling of a flexible flow shop that consists of several processing stages in series separated by finite intermediate buffers and one or more parallel machines at each stage. Then, the basic mixed-integer programming formulations have been enhanced to model blocking scheduling with alternative processing routes where for each product a set of routes is available for processing. Besides, a reentrant flow line where a product visits a set of stages more than once is also considered. The objective is to determine a production schedule that minimizes makespan considering several product types. In the same way as Ribas (2017), the mathematical model of this paper becomes a good example for us to construct one of our mixed-integer programming formulations in our study.

Grabowski and Pempera (2000) focus on a real-life problem in the production of concrete blocks in a factory of the building industry. Aiming at minimizing makespan of the flexible flow shop with no wait constraint, an approximation algorithm based on the tabu search approach is applied.

Azizoğlu et al. (2001) consider flexible flow shop scheduling problem with n jobs on w stages, each stage including several parallel identical machines with the objective function as total flow time minimization. A Branch and Bound algorithm is developed to find the optimal schedule and three heuristics are proposed to find an initial solution for the Branch and Bound algorithm.

Soewandi and Elmaghraby (2001) focus on the problem of sequencing *n* jobs in a 3-stage flexible flow shop with multiple identical machines at each stage. Several lower bounds and heuristic procedures are developed for minimizing makespan.

Gupta et al. (2002) consider a flexible flow shop scheduling problem with release dates and controllable processing times. Constructive heuristic algorithms based on job insertion techniques and iterative heuristic algorithms based on local search are proposed to reach an optimal solution which is the sum of earliness and tardiness penalties, weighted completion times of jobs and the costs of due date assignments.

Wardono and Fathi (2004) investigate the hybrid flow shop scheduling problem that has *N* jobs on parallel machines in *L* successive stages with limited buffer capacities between stages. A tabu search algorithm is developed to get the schedule that would minimize the makespan.

Kyparisis and Koulamas (2006) consider the multistage flexible flow shop scheduling problem with uniform parallel machines at each stage and the objective of minimizing makespan. A general class of heuristics for this strongly NP-hard problem that extends several well-known heuristics for the corresponding embedded serial flow shop problem are developed.

Quadt and Khun (2007) present taxonomy for flexible flow shop scheduling procedures that have parallel machines at some or all production stages. The taxonomy groups the procedures according to their general solution approach such as optimal procedures and heuristic procedures. Besides, a list of studies is given taking account of objective, solution approach and parallel machines characteristic.

Ruiz et al. (2008) present a mixed-integer mathematical model and six heuristic algorithms for flexible flow shop scheduling problem where sequence-dependent setup times and makespan minimization as an objective function are considered, and jobs might skip stages. The mixed-integer programming and the proposed heuristics are tested against a comprehensive benchmark and the results evaluated by advanced statistical tools.

Tavakkoli-Moghaddam et al. (2009) introduce a memetic algorithm combined with a novel local search engine, namely, nested variable neighborhood search to solve the flexible flow line scheduling problem with processor blocking and intermediate buffers. The flexible flow line consists of several processing stages in series with each stage having one or more identical parallel machines. The objective is to determine a production schedule for all types of products to be completed in a minimum time (minimize makespan).

Ruiz and Vázquez-Rodríguez (2010) present a literature review on the subject of hybrid flow shop scheduling problem by grouping studies according to their solution methods such as exact algorithm, heuristic and metaheuristic methods. Several variants of the HFS problem, each in turn considering different assumptions, constraints and objective functions are discussed and reviewed. In the end, research opportunities in HFS scheduling are discussed.

Engin et al. (2011) study flexible flow shop scheduling problem with makespan minimization. An efficient genetic algorithm with Neighborhood Based Mutation is proposed as a solution method. Considering the comparison of the proposed heuristic algorithm with the genetic algorithm and parallel greedy heuristic algorithm, the proposed efficient genetic algorithm performs better than the genetic algorithm and the parallel greedy heuristic algorithm.

Liao et al. (2012) propose a metaheuristic algorithm, namely, practical swarm optimization algorithm, for the flexible flow shop scheduling problem with makespan minimization as an objective function. Besides, the bottleneck stage is fully exploited by the proposed bottleneck heuristic and it is able to escape from local optima by the proposed simulated annealing heuristic. According to the comparison of heuristics, the proposed algorithm performs better than others.

Almeder and Hartl (2013) present a scheduling problem of a real-world production process in the metal-working industry, which is a 2-stage flexible flow shop problem with limited buffer between stages, aiming to minimize expected average completion time. The variable neighborhood search method is used to obtain the result.

Marichelvam et al. (2014) present a recently developed cuckoo search metaheuristic algorithm for flexible flow shop scheduling problem with makespan minimization as an objective. The NEH algorithm is used to acquire a good initial solution for the cuckoo search algorithm. The proposed heuristic algorithm outperforms many other metaheuristics.

Li and Pan (2015) focus on a novel hybrid algorithm that consists of artificial bee colony and tabu search approaches for flexible flow shop scheduling problem with limited buffer which has an objective function of makespan minimization. As a result of computational study with sets of large instances, the proposed algorithm performs better than the other existing heuristic algorithms in the literature.

Dios et al. (2018) address the hybrid flow shop scheduling problem for makespan minimization with special cases where there are missing operations. A set of heuristic approaches that capture some special features of the missing operations are proposed. Then these heuristic algorithms are compared with already existing heuristics for the classical hybrid flow shop, and the hybrid flow shop problem with missing operations.

CHAPTER 4

SOLUTION APPROACHES

In this chapter, we present solution alternatives for our problem. Since our problem is similar to both flexible flow shop scheduling and parallel flow shop scheduling in different aspects, we intend to approach the problem considering aspects of both. Hence, we develop two Mixed Integer Linear Programming (MILP) models in an attempt to obtain the optimum solution. One of them is a flexible flow shop (FFS) based model, while the other one is a parallel flow shop (PFS) based model. Furthermore, we propose two heuristic algorithms to get a near-optimal solution, if not the optimum, in a faster way. Finally, in order to evaluate our heuristic solution methods against the optimum and other methods when the optimum is not attained by the MILP models, we resort to some well-known dispatching rules, namely Shortest Processing Time (SPT), and Longest Processing Time (LPT) rules.

4.1. Flexible Flow Shop Based MILP Model

The first mathematical model is developed with FFS scheduling approach and can be considered an extension of the MILP model developed for flexible flow shop scheduling problem with limited intermediate buffers proposed by Sawik (2000). In the scope of Sawik's study, the flexible flow shop consists of n stages in series, where each stage has m identical parallel machines and finite intermediate buffers between stages. The shop produces k types of different products in a mixed order and the sequence of products is the same for all stages. Besides, each product must be processed by at most one machine at each stage. When an operation of a product is completed on a machine and the intermediate buffer is occupied (it means the next machine is already blocked), then the product remains on the current machine and blocks it until the next buffer becomes available. The objective is to determine a production schedule for all products so as to complete all products in minimum time possible.

In addition to Sawik (2000)'s study, we use additional constraints that prevent products from crossing between the two flow shops; that means the products must progress on and leave from the flow shop where they enter at the beginning. Furthermore, although there is not any restriction about the number of parallel machines at any stage, we limit it by two machines (m=2). In other words, some stages may contain two workstations and some others may have only one. When there is just one workstation at a stage, we call it a "common workstation" which serves both flow shops. Lastly, we restrict the capacity of the intermediate buffer area in front of each machine by one product, meaning that totally a buffer area for two products are kept between any two consecutive stages, one buffer area for each flow shop.

Parameters, Sets and Indices

k: number of products

n: number of processing stages

m: number of parallel workstations at stage i; $m = |J_i|$

j: workstation index at stage i, where $j \in J_i = \{1, . . . , m\}$

p: product index, $p \in P = \{1, . . . , k\}$

i: stage index, $i \in I = \{1, . . . , n\}$

$t_{i,p}$: processing time of product p at stage i

Q: a large number, not less than schedule length

Decision Variables

C_{max}: schedule length

$c_{i,p}$: completion time of product p at stage i

$d_{i,p}$: departure time of product p from stage i

$$X_{i,j,p} : \begin{cases} 1, \text{if product } p \text{ is assigned to station } j \in J_i \text{ at stage } i \in I \\ 0, \text{otherwise} \end{cases}$$

$$Y_{p,r} : \begin{cases} 1, \text{if product } p \text{ precedes product } r \\ 0, \text{otherwise} \end{cases}$$

MILP Model

Objective function:

$$\text{Min } C_{max} \tag{1}$$

The objective function (1) represents the schedule length (total completion time) of all jobs (products) to be minimized.

Constraints:

$$\sum_{j \in J_i} X_{i,j,p} = 1 \qquad i \in I \; ; \; p \in P \; ; \; |J_i| \geq 1 \tag{2}$$

Constraint set (2) ensures that each product should be assigned to exactly one workstation at every stage.

$$X_{i,1,p} + X_{i+1,2,p} = 1 \qquad i \in I \; ; \; p \in P \tag{3}$$

$$X_{i,2,p} + X_{i+1,1,p} = 1 \qquad i \in I \; ; \; p \in P \tag{4}$$

Constraint sets (3) and (4), which are added in our proposed mathematical model, prevent the crossing of products (jobs) between the two flow shops; thanks to constraint set (3) the products cannot pass from flow shop 1 to flow shop 2; likewise, constraint set (4) prevents crossing from flow shop 2 to flow shop 1.

$$\sum_{p \in P} t_{i,p} \; X_{i,j,p} \leq \sum_{p \in P} \frac{t_{i,p}}{|J_i|} + min(t_{i,p})_{p \in P} \qquad i \in I \; ; \; j \in J_i \; ; \; |J_i| \geq 1 \tag{5}$$

Constraint set (5) enables the workload to be balanced among all parallel workstations at every stage.

$$c_{1,p} \geq t_{1,p} \qquad p \in P \qquad (6)$$

$$c_{i,p} - c_{i-1,p} \geq t_{i,p} \qquad i \in I\ ;\ p \in P\ ;\ i > 1 \qquad (7)$$

Constraint set (6) ensures that each product is processed at the first stage, and constraint set (7) guarantees that it is then processed at all downstream stages.

$$c_{i,p} + Q\ \ Y_{p,r} \geq d_{i,r} + t_{i,p} \qquad i \in I\ ; r,p \in P\ ;\ p < r\ ;\ |J_i| = 1 \qquad (8)$$

$$c_{i,r} + Q\ \left(1 - Y_{p,r}\right) \geq d_{i,p} + t_{i,r} \qquad i \in I\ ;\ r,p \in P\ ;\ p < r\ ;\ |J_i| = 1 \qquad (9)$$

Constraint sets (8) and (9) indicate the precedence relationships among products at the stage that contains a single workstation.

$$c_{i,p} + Q\ \left(2 + Y_{r,p} - X_{i,j,p} - X_{i,j,r}\right) \geq d_{i,r} + t_{i,p} \qquad i \in I\ ; j \in J_i\ ;\ r,p \in P\ ;\ p < r\ ;\ |J_i| > 1 \qquad (10)$$

$$c_{i,r} + Q\ \ \left(3 - Y_{r,p} - X_{i,j,p} - X_{i,j,r}\right) \geq d_{i,p} + t_{i,p} \qquad i \in I\ ; j \in J_i\ ;\ r,p \in P\ ;\ p < r\ ;\ |J_i| > 1 \qquad (11)$$

Similar to constraint sets (8) and (9), constraint sets (10) and (11) indicate the precedence relationships among products. However, these constraints are active at the stages which have parallel workstations.

$$c_{i,p} = d_{i-1,p} + t_{i,p} \qquad i \in I\ ;\ p \in P\ ;\ i > 1 \qquad (12)$$

Constraint set (12) guarantees that as soon as a product leaves a workstation or buffer area, its next operation starts immediately at the workstation of the next stage.

$$c_{n,p} = d_{n,p} \qquad p \in P \qquad (13)$$

Constraint set (13) ensures that each product leaves the shop once its operation is completed at the last stage.

$$c_{n,p} \leq C_{max} \qquad p \in P \qquad (14)$$

$$C_{max} \geq \sum_{p \in P} \frac{t_{i,p}}{|J_i|} + \sum_{h \in I: h \neq i} min(t_{h,p})_{p \in P} \qquad i \in I \qquad (15)$$

While constraint set (14) defines the maximum completion time, constraint set (15) imposes a lower bound on it.

$$X_{i,j,p} = 0 \qquad i \in I \; ; \; j \in J_i \; ; \; p \in P \; ; \; |J_i| = 1 \qquad (16)$$

$$Y_{r,p} = 0 \qquad r, p \in P \; ; \; p \geq r \qquad (17)$$

Constraint sets (16) and (17) eliminate some certain decision variables by setting them to zero. At stages where there is single workstation, the decision variables $X_{i,j,p}$ take on zero value for the second flow shop. On the other hand, the decision variables $Y_{r,p}$ take on zero value when product *r* precedes product *p*.

$$c_{i,p} \geq 0 \qquad i \in I \; ; \; p \in P \qquad (18)$$

$$d_{i,p} \geq 0 \qquad i \in I \; ; \; p \in P \qquad (19)$$

$$C_{max} \geq 0 \qquad (20)$$

$$X_{i,j,p} \in \{0, 1\} \qquad i \in I \; ; \; j \in J_i \; ; \; p \in P \qquad (21)$$

$$Y_{r,p} \in \{0, 1\} \qquad r, p \in P \qquad (22)$$

While constraint sets (18), (19), and (20) indicate that the related decision variables are to be nonnegative, constraint sets (21) and (22) express that the related decision variables are to be binary.

As it is obvious from the above formulation, the FFS based MILP model consists of numerous decision variables and constraints as follows (taking $|J|$=$|J_i|$):

- Number of binary variables: $|I| \times |J| \times |P| + |P|^2$
- Number of continuous variables: $2 \times |I| \times |P| + 1$
- Number of constraints: $|I| \times |P| \times \big((|J| + 1) \times (|P| + 1) + 4\big) \div 2 + |P| \times (4\,|I| + |P| + 3) \div 2 + |I| \times (|J| + 1) + 1$

This MILP model is a general formulation for flexible flow shop scheduling with limited buffers and can be customized. For instance, if $|J_i|$ is assumed to be one, for all $i \in I$, then our model reduces to a model for flow shop scheduling with limited buffer. Likewise, if $t_{i,p}$ is a positive value for all i and p, then our mathematical model reduces to a model for flexible flow shop scheduling with no buffers.

4.2. Parallel Flow Shop Based MILP Model

The other mathematical model in this study is derived from the parallel flow shop scheduling problem, and can be considered to be an extension of the MILP model for parallel blocking flow shop scheduling problem proposed by Ribas (2017). In the scope of Ribas (2017)'s study, the *F* identical parallel flow shops, each consisting of a series of *m* machines, are considered. The flow shops produce *n* different products in a mixed order, and the sequence of products is the same for all machines. There is no buffer area between machines, and when an operation of a product is completed on a machine, and the next machine is occupied, then the product remains on that current machine and blocks it until the next machine becomes available. Likewise our problem, Ribas (2017)'s model does not allow products to cross between parallel flow shops. This means that products must progress on and leave from the flow shop which they enter at the beginning. The objective is to determine a production schedule for all products so as to complete the products in minimum time.

Different from Ribas (2017)'s study, we use additional constraints to adapt it to our case. As we implement on the FFS based MILP model, we limit the number of parallel flow shops by two (s=2) although there is not any restriction about it in Ribas (2017)'s model. This means that some stages may contain two workstations, which corresponds to one workstation at that stage of each flow shop and some other stages may have only one workstation. When there is just one workstation at a stage, we call it a common workstation which serves both flow shops. Furthermore, we prefer to use buffer areas between machines at each flow shop and restrict the capacity of each intermediate buffer area by one product.

Parameters, Sets and Indices

n: number of products (jobs)

m: number of machines (stages, workstations)

s: number of flow shops (s=2 in our case)

p: number of positions

j: job index, $j \in J=\{1, \dots, n\}$

i: machine index, $i \in I=\{1, \dots, m\}$

f: flow shop index, $f \in F=\{1, \dots, s\}$

k: position index, $k \in K=\{1, \dots, n\}$

$t_{i,j}$: processing time of job j at machine i

H: set of common workstations

$|I_i|$: number of workstations at both flow shops for stage i

Decision Variables

C_{max}: schedule length

$d_{i,k,f}$: departure time of a job that occupies position k at machine i of flow shop f

$X_{k,j,f}$: $\begin{cases} 1, \text{if job } j \text{ is assigned to position } k \in K \text{ at flow shop } f \in F \\ 0, \text{otherwise} \end{cases}$

MILP Model

Objective function:

$$\text{Min } C_{max} \quad (1)$$

The objective function (1) represents the schedule length (total completion time) to be minimized.

Constraints:

$$\sum_{f=1}^{2} \sum_{k=1}^{n} X_{k,j,f} = 1 \qquad j \in J \quad (2)$$

Constraint set (2) ensures that each job should be assigned to exactly one position only at one flow shop.

$$\sum_{j=1}^{n} X_{k,j,f} \leq 1 \qquad k \in K \; ; \; f \in F \quad (3)$$

Constraint set (3) ensures that at most one job can be allocated to each position at a flow shop.

$$d_{1,k,f} \geq d_{1,k-1,f} + \sum_{j=1}^{n} X_{k,j,f} \quad t_{1,j} \qquad k \in K \, ; k \neq 1 \; ; \; f \in F \quad (4)$$

$$d_{1,1,f} \geq \sum_{j=1}^{n} X_{1,j,f} \quad t_{1,j} \qquad f \in F \quad (5)$$

Constraint sets (4) and (5) define the departure time of the job which occupies position *k* in the first machine at flow shop *f*.

$$d_{i,k,f} \geq d_{i-1,k,f} + \sum_{j=1}^{n} X_{k,j,f} \quad t_{i,j} \qquad i \in I \; ; \; i \neq 1 \; ; \; k \in K \; ; \; f \in F \quad (6)$$

Constraint set (6) indicates the relationship between the departure times of each job in two consecutive machines at the assigned flow shop.

$$d_{i,k,1} \geq d_{i,k-1,2} + \sum_{j=1}^{n} X_{k,j,1} \quad t_{i,j} \qquad i \in H\ ;\ k \in K\ ;\ k \neq 1 \tag{7}$$

$$d_{i,k,2} \geq d_{i,k,1} + \sum_{j=1}^{n} X_{k,j,2} \quad t_{i,j} \qquad i \in H\ ;\ k \in K \tag{8}$$

Constraint sets (7) and (8), which are added in our proposed mathematical model, organize the order of jobs that enter the common workstations.

$$d_{i,k,f} \geq d_{i+1,k-1,f} \qquad i \in I\ ;\ i \neq m\ ;\ k \in K\ ;\ k \neq 1\ ;\ f \in F \tag{9}$$

Constraint set (9) determines the departure time of a job under the blocking conditions by considering that the next machine has to be available.

$$C_{max} \geq d_{m,n,f} \qquad f \in F \tag{10}$$

Constraint set (10) defines the makespan.

$$C_{max} \geq \sum_{j \in J} \frac{t_{i,j}}{|I_i|} + \sum_{h \in I: h \neq i} min(t_{h,j})_{j \in J} \qquad i \in I \tag{11}$$

Constraint set (11) imposes a lower bound on makespan that is proposed by Sawik (2000) and adapted here.

$$X_{k,j,f} \in \{0, 1\} \qquad k \in K\ ;\ j \in J\ ;\ f \in F \tag{12}$$

$$d_{i,k,f} \geq 0 \qquad i \in I\ ;\ k \in K\ ;\ f \in F \tag{13}$$

$$C_{max} \geq 0 \tag{14}$$

While constraint set (12) forces the related decision variables to be binary, constraint sets (13) and (14) express that the related decision variables are continuous and cannot take on a negative value.

The PFS based MILP model consists of numerous decision variables and constraints as well as shown below:

- Number of binary variables: $|K| \times |J| \times |F|$

- Number of continuous variables: $|I| \times |K| \times |F| + 1$
- Number of constraints: $|K| \times |F| \times (I + 1) + 2 \times |I| \times |K| + |I - 1| \times |K - 1| \times |F| + |J| + |F| + 1$

Unlike our study, Ribas (2017) prefers to omit buffer areas. However, in our mathematical model, we suppose the buffer areas as a machine with the processing times ($t_{i,j}$) being equal to zero. If the processing times are taken as positive values, then our mathematical model reduces to a model for parallel flow shop scheduling with common workstations and no buffer.

4.3. Heuristic Approaches

4.3.1. Proposed Heuristic Approaches

The proposed heuristic algorithms in this study can be considered as an extension of the NEH Algorithm (Nawaz et al. 1983). The NEH Algorithm, which is considered as one of the best simple and constructive heuristics for the makespan minimization, was developed by Nawaz, Enscore, and Ham in 1983. Since then, several studies have accepted the NEH algorithm as a reference, and thus, either the NEH algorithm was used directly in the studies or the efficiency of the newly developed heuristic methods within the scope of the studies was evaluated by comparing them with the NEH algorithm. In this sense, certain studies claim that the NEH algorithm is a better method than the other heuristics. For example, Taillard (1990) makes a comparison of the classical heuristics including Gupta's algorithm, Johnson's algorithm, Rapid Access (RA) procedure, Palmer's algorithm, CDS algorithm, and NEH algorithm, and finds out that NEH algorithm gives the best solution as a result. Likewise, Semanco and Modrak (2012) propose a constructive heuristic approach named MOD and compare it with the Gupta's algorithm, Palmer's algorithm, CDS algorithm and NEH algorithm, and observes that NEH algorithm outperforms all other heuristics. Furthermore, Ruiz et al. (2008) compare five dispatching rules and NEH algorithm.

As a result of their comparison, NEH algorithm is vastly superior to the dispatching rules.

The NEH algorithm proceeds in an iterative manner based on the logic that jobs with high total processing times must be scheduled as early as possible in the sequence. The steps that explain the working principle of the algorithm are as follows:

Step 1: Find the total processing time for each job *p* using the expression below:

$$\sum_{i=1}^{n} t_{i,p} \qquad \forall p \in P$$

Step 2: Sort the jobs in the non-increasing order of their total processing times.

Step 3: Select the first two jobs that have the highest total processing times, and calculate C_{max} of the two partial sequences by interchanging the positions of the two jobs like J1-J2 and J2-J1. Then eliminate the sequence having the higher value of C_{max}, and determine the sequence with the lower C_{max} value as the current sequence, and continue to *Step* 4 with this sequence.

Step 4: Select the next job from the list and place it in the possible position in the current sequence. Calculate C_{max} of all alternatives and determine the sequence with the lower C_{max} value as the new current sequence. For example, in the case of J2-J1 as the current sequence from *Step* 2, we have three alternative sequences to consider, J2-J1-J3, J2-J3-J1, and J3-J2-J1. If J2-J3-J1 sequence has the lower C_{max} among them, we state this sequence as the new current sequence and discard the other alternatives.

Step 5: If there are unordered jobs left in the list, return to *Step* 4. Otherwise, stop here.

In this study, we implement the NEH algorithm with some modifications. We carry out *Step* 1 and *Step* 2 in the same way as the NEH algorithm. Since we have two parallel flow shops, there are certain differences in applying *Step* 3. After sorting the jobs in non-increasing order, we select the first two jobs that have higher total processing times and assign the highest one to flow shop 1 and the second highest one to flow shop 2. For example, J1 is assigned to flow shop 1 and J2 to flow shop 2. After that, we select the next job from the list and try all possible alternatives as follows:

- J1-J3 at flow shop 1 & J2 at flow shop 2
- J3-J1 at flow shop 1 & J2 at flow shop 2
- J1 at flow shop 1 & J2-J3 at flow shop 2
- J1 at flow shop 1 & J3-J2 at flow shop 2

For all alternatives above, we calculate C_{max} and select the sequence which has the lowest C_{max} among them. From this time on, this sequence becomes the current sequence, and we continue to the next step with this current sequence. *Step* 4 repeats until there are no unordered jobs left in the list. In this way, we implement the NEH algorithm to our problem, that is parallel flow shops with common workstations.

Moreover, as it is seen above, at the beginning of the NEH algorithm, jobs are sorted in a non-increasing order. In our study, we sort the jobs in a non-decreasing order as well as a non-increasing order. Thus, we have the opportunity to examine more alternatives to attain a better solution. In the rest of this study, we call the proposed solution method in which we sort the jobs in a non-decreasing order as SPT&NEH Algorithm and the solution method in which we sort the jobs in a non-increasing order as LPT&NEH Algorithm.

The classic NEH Algorithm has $(n \times (n+1))/2$ alternatives where n is the number of jobs, which clearly states that the complexity of this algorithm is $O(n^2)$. On the other hand, the proposed NEH Algorithm has $(n \times (n+3))/2$ alternatives, which obviously shows us that the complexity of the proposed NEH Algorithm is $O(n^2)$ as well.

Now, let us look at the simple example problem below to figure out clearly how our algorithm works.

Example:

Let there be two parallel flow shops, FS1 and FS2, which consist of three stages (n=3). While in the first stage there are two workstations, second and third stages have a single workstation meaning the common workstation. There are buffer areas between consecutive stages each with one product capacity (Figure 4.1).

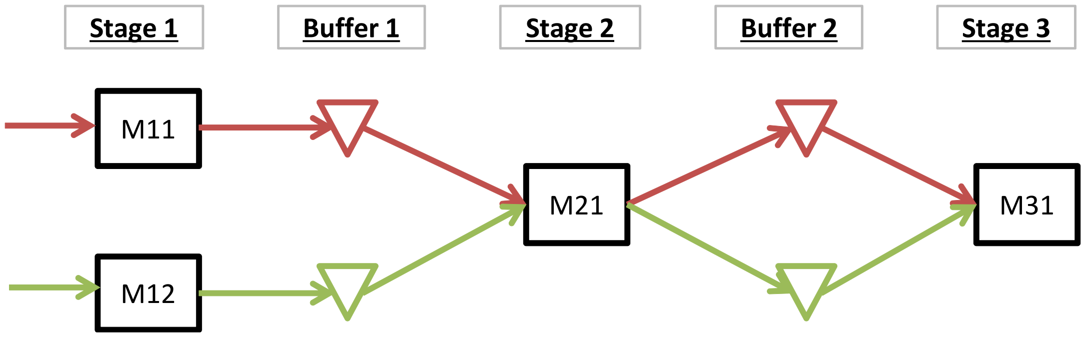

Figure 4.1 An Example for a Parallel Flow Shop

We have five products to be processed, P = {J1, J2, J3, J4, J5}, and their processing times are as shown in Table 4.1.

First, we calculate the total processing time of each job and then sort them in a non-increasing order according to their total processing time as below for LPT&NEH Algorithm.

Table 4.1 Processing Times of Jobs, $t_{i,j}$ (minutes)

Stages\Jobs	J1	J2	J3	J4	J5
S1*	43	55	47	38	32
B1	0	0	0	0	0
S2	29	37	30	15	22
B2	0	0	0	0	0
S3	0	0	21	18	26
Total Processing Time	72	92	98	71	80

*S# : stage # and B#: buffer #

Table 4.2 Non-increasing Order of Jobs by Total Processing Time

Jobs	J3	J2	J5	J1	J4
Total Processing time	98	92	80	72	71

Afterwards, we assign J3 to FS1 and J2 to FS2, and then we try all alternatives for J5 with makespan calculations as follows:

- J3-J5 at FS1 & J2 at FS2 → C_{max} = 162
- J5-J3 at FS1 & J2 at FS2 → C_{max} = 143
- J3 at FS1 & J2-J5 at FS2 → C_{max} = 162
- J3 at FS1 & J5-J2 at FS2 → C_{max} = 124 → Current sequence

As a result of the first iteration we find the sequence J3 at FS1 and J5-J2 at FS2 with the minimum C_{max}=124 and we set it as the current sequence. Now we continue to the next iteration with the current sequence, and try all alternatives for J1.

- J3-J1 at FS1 & J5-J2 at FS2 → C_{max} = 153 → Current sequence
- J1-J3 at FS1 & J5-J2 at FS2 → C_{max} = 175
- J3 at FS1 & J5-J2-J1 at FS2 → C_{max} = 159
- J3 at FS1 & J5-J1-J2 at FS2 → C_{max} = 167
- J3 at FS1 & J1-J5-J2 at FS2 → C_{max} = 167

At the end of the second iteration we reach the sequence J3-J1 at FS1 and J5-J2 at FS2 with the minimum C_{max}=153 and this sequence becomes our new current sequence. Finally, we add J4 to an appropriate position.

- J3-J1-J4 at FS1 & J5-J2 at FS2 → C_{max} = 186
- J3-J4-J1 at FS1 & J5-J2 at FS2 → C_{max} = 166
- J4-J3-J1 at FS1 & J5-J2 at FS2 → C_{max} = 171
- J3-J1 at FS1 & J5-J2-J4 at FS2 → C_{max} = 186
- J3-J1 at FS1 & J5-J4-J2 at FS2 → C_{max} = 165 →Final sequence
- J3-J1 at FS1 & J4-J5-J2 at FS2 → C_{max} = 171

As a conclusion of all iterations, we find the optimum sequence as J3-J1 at FS1 and J5-J4-J2 at FS2 with C_{max}=165.

Now, we follow the same steps by sorting the jobs in a non-decreasing order regarding their total processing times for SPT&NEH Algorithm (Table 4.3).

Table 4.3 Non-decreasing Order of Jobs by Total Processing Time

jobs	J4	J1	J5	J2	J3
Total Processing Time	71	72	80	92	98

1st Iteration:

- J4-J5 at FS1 & J1 at FS2 → C_{max} = 130
- J5-J4 at FS1 & J1 at FS2 → C_{max} = 116
- J4 at FS1 & J1-J5 at FS2 → C_{max} = 130
- J4 at FS1 & J5-J1 at FS2 → C_{max} = 104 → Current sequence

2nd Iteration:

- J4-J2 at FS1 & J5-J1 at FS2 → C_{max} = 141 → Current sequence
- J2-J4 at FS1 & J5-J1 at FS2 → C_{max} = 154
- J4 at FS1 & J5-J1-J2 at FS2 → C_{max} = 167
- J4 at FS1 & J5-J2-J1 at FS2 → C_{max} = 159
- J4 at FS1 & J2-J5-J1 at FS2 → C_{max} = 159

3rd Iteration:

- J4-J2-J3 at FS1 & J5-J1 at FS2 → C_{max} = 192
- J4-J3-J2 at FS1 & J5-J1 at FS2 → C_{max} = 177
- J3-J4-J2 at FS1 & J5-J1 at FS2 → C_{max} = 177

- J4-J2 at FS1 & J5-J1-J3 at FS2 → C_{max} = 192
- J4-J2 at FS1 & J5-J3-J1 at FS2 → C_{max} = 175
- J4-J2 at FS1 & J3-J5-J1 at FS2 → C_{max} = 171 → Final sequence

As a conclusion of the example, we find the sequence of J3-J1 at FS1 & J5-J4-J2 at FS2 with C_{max}=165 by sorting the jobs in the non-increasing order, and the sequence of J4-J2 at FS1 & J3-J5-J1 at FS2 with C_{max}=171 by sorting the jobs in the non-decreasing order. Since we aim to minimize the makespan, we determine the exact sequence as J3-J1 at FS1 & J5-J4-J2 at FS2 with C_{max}=165. The Gantt chart of the sequence can be seen below in Figure 4.2.

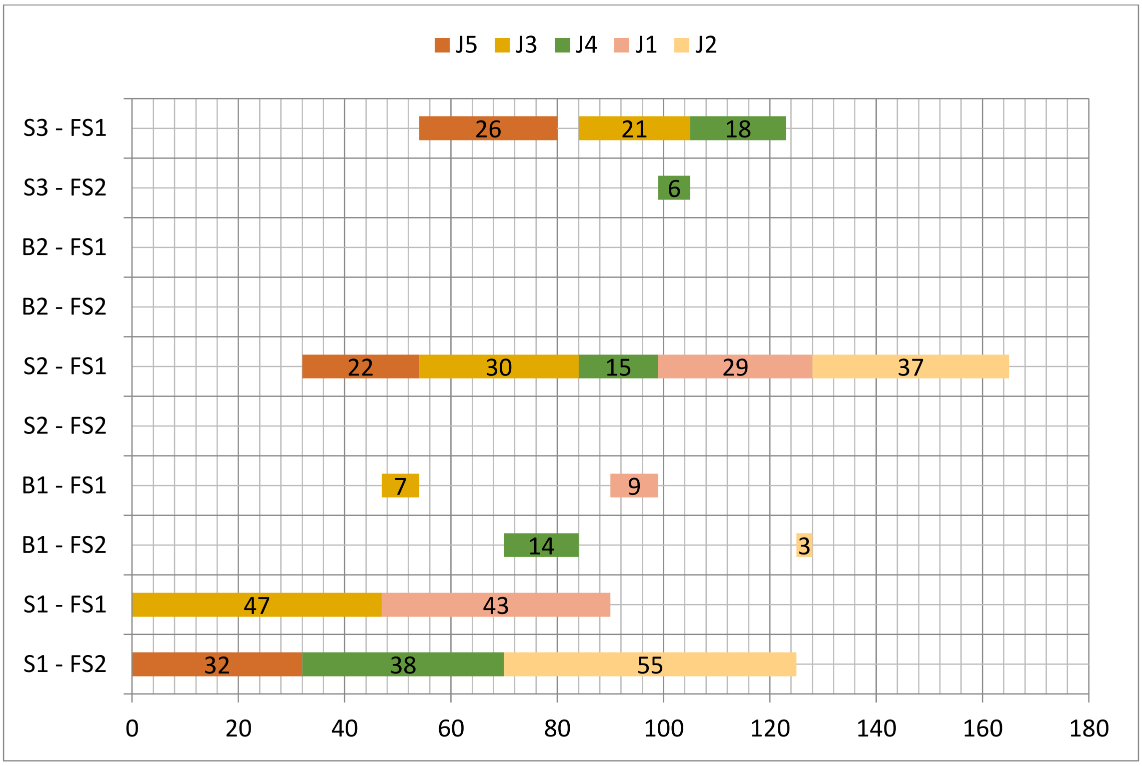

Figure 4.2 Gantt Chart of the Best Sequence for the Example Problem

4.3.2. Dispatching Rules

Dispatching rules are classified as construction heuristics and preferred to be used both in practice and numerous studies in the literature. Dispatching rules prioritize all jobs at the beginning, and then, whenever a machine becomes available, dispatching rules select the job with the highest priority to be processed. Dispatching rules consist of several types of rules, and the two most commonly used of them are the Shortest Processing Time (SPT) and Longest Processing Time (LPT) rules. The working principle of these rules can be explained as below:

Shortest Processing Time (SPT)

Step 1: Find the total processing time for each job using the expression below:

$$\sum_{i=1}^{n} t_{i,p} \qquad \forall p \in P$$

Step 2: Sort the jobs in the non-decreasing order of their total processing times.

Step 3: Select the first job that has the shortest total processing time to be processed.

Step 4: Select the next job from the list when the machine becomes available.

Step 5: If there are jobs left in the list, return to *Step* 4. Otherwise, stop.

In this study, since we have two parallel flow shops, we implement the SPT rule with some modifications. We follow the first two steps in the same way, but in *Step* 3, we select the first two jobs from the list and assign the shortest one to flow shop 1 and the second shortest to flow shop 2. Then in *Step* 4, we select the third job from the list and assign it to the flow shop, which becomes available earlier, and finally, this step is repeated until there is no job left in the list.

The complexity of the modified SPT rule used in this study is the same with the complexity of the classic SPT rule which is $O(n \times \log(n))$.

We address the same example given above in the proposed heuristic section (Table 4.4):

Table 4.4 SPT Order of Jobs by Total Processing Time

Jobs	J4	J1	J5	J2	J3
Total Processing Time	71	72	80	92	98

We sort the jobs as J4-J1-J5-J2-J3, and then assign the jobs to the two flow shops. By this way, the result is found as below:

- J4-J5-J3 at FS1 & J1-J2 at FS2 → C_{max}=192 → Final sequence.

As a reminder, we have found C_{max}=171 with SPT&NEH method and C_{max}=165 with LPT&NEH method.

Longest Processing Time (LPT)

This rule is almost the same as the SPT rule. The only difference is that in *Step* 2 of the SPT rule, we sort the jobs in the non-decreasing order of their total processing times, while we sort the jobs in the non-increasing order of their total processing times with the LPT rule. Thus, the job with the longest processing time gains the chance to enter the flow shop as the first job.

Similar to the SPT rule, the complexity of the modified LPT rule is $O(n \times \log(n))$.

We examine the same example above (Table 4.5).

Table 4.5 LPT Order of Jobs by Total Processing Time

Jobs	J3	J2	J5	J1	J4
Total Processing Time	98	92	80	72	71

We order the jobs as J3-J3-J5-J1-J4, and find the result as shown:

- J3-J5-J4 at FS1 & J2-J1 at FS2 → C_{max}=198 → Final sequence.

As a reminder, we have found C_{max}=171 with SPT&NEH method, C_{max}=165 with LPT&NEH method and C_{max}=192 with the SPT rule.

CHAPTER 5

COMPUTATIONAL STUDY

In this chapter, we examine our solution approaches presented in the previous chapter through several experiments. First, we run both MILPs for all experiments, compare them with each other and determine the better one. After that, we analyze the four heuristic approaches, namely SPT&NEH, LPT&NEH, SPT, LPT, and find the best of them for each experiment. Finally, we compare the results of heuristics against the better result of the two MILPs.

Comparing the proposed mathematical model and the proposed heuristic approaches with the well-known dispatching rules, SPT and LPT, is vital in order to carry out a better evaluation of the proposed solution approaches. Thus, we can observe both the effectiveness and efficiency of our proposed heuristics against SPT and LPT rules.

The primary purpose of the computational study is to observe how our proposed heuristic approach -the modified NEH algorithm- behaves in our real-life case and how it reacts to the number of jobs, number of operators and shop design. While observing this, we focus on three main criteria:

1. Solution quality: closeness to the optimum solution.
2. Running time: computational time for a certain approach in a specific shop design.
3. Shop design: effectiveness of using common workstations in parallel flow shops.

Our expectation from the proposed heuristic algorithm is to find out a suitable solution that can be an optimal solution or a good near-optimal solution. We observe the solution quality of our proposed heuristics by the “gap” value, which shows the distance between the solution of the MILP and the solution of the proposed heuristic by percentage as follows:

$$GAP = \frac{C_{max} - C^*_{max}}{C^*_{max}} \times 100\ \%$$

C_{max} value is obtained by the heuristic approach and C^*_{max} value is the one found by MILP.

In addition, it is vital for us how fast the proposed heuristic algorithm reaches the result, as well as reaching a good solution, and also it should be within an acceptable time interval.

Likewise, the results of the solution approaches for each shop design alternative are very crucial to evaluate the performance of using common workstations in the parallel flow shops. The positioning of the common workstations is as important as the number of common workstations placed, and these two criteria directly affect the results of the experiments. We also compare and discuss the result of shop design alternatives.

In order to obtain more detailed results and to see the effectiveness of the proposed solution approaches, we categorize the numerical experiments by two criteria: the total number of operators working on the two flow shops and the total number of jobs processed on the two flow shops. While we consider 4, 5, 6, 10, and 15 as the total number of operators that can work at the two flow shops, we use 15, 25, and 50 jobs to be processed at the two flow shops in our experiments. Thus, we have the opportunity to test our proposed heuristic approaches for different problem sizes.

On the other hand, depending on the number of operators, different shop design alternatives arise. We analyze all of the alternatives that result with 4, 5, and 6 operators, namely, 5, 8, and 13 shop design alternatives, respectively. However, for 10 and 15 operators, we focus on just five random shop alternatives due to the very large number of shop design alternatives that result with 10 operators and more. The shop design alternatives for each level of workforce (number of operators) are given in Table 5.1.

Furthermore, to obtain a more detailed result, we consider five different experiments for each case. The summary of the computational study is presented in Table 5.2.

Table 5.1 Shop Design Alternatives

# of Operators	**4**	**5**	**6**	**10**	**15**
Shop Design Alternatives	2-2	2-2-1	2-2-2	2-2-2-2-2	2-2-2-2-2-2-2-1
	2-1-1	2-1-2	2-2-1-1	2-1-2-1-2-2	2-2-1-1-2-1-1-2-1-1-1
	1-2-1	1-2-2	2-1-2-1	1-1-2-1-2-2-1	1-1-1-1-2-2-2-2-1-2
	1-1-2	2-1-1-1	2-1-1-2	1-1-2-1-1-2-1-1	1-2-2-1-1-1-1-1-1-2-2
	1-1-1-1	1-2-1-1	1-2-2-1	1-1-1-1-1-1-1-1-1-1	1-1-1-1-1-1-1-1-1-1-1-1-1-1
	-	1-1-2-1	1-2-1-2	-	-
	-	1-1-1-2	1-1-2-2	-	-
	-	1-1-1-1-1	2-1-1-1-1	-	-
	-	-	1-2-1-1-1	-	-
	-	-	1-1-2-1-1	-	-
	-	-	1-1-1-2-1	-	-
	-	-	1-1-1-1-2	-	-
	-	-	1-1-1-1-1-1	-	-
Total # of Shop Design Alternatives	5	8	13	5	5

Table 5.2 Summary of the Computational Study

# of Operators	# of Jobs	# of Shop Design Alternatives	# of Experiments	# of Problem Instances
4	15	5	5	25
	25	5	5	25
	50	5	5	25
5	15	8	5	40
	25	8	5	40
	50	8	5	40
6	15	13	5	65
	25	13	5	65
	50	13	5	65
10	15	5	5	25
	25	5	5	25
	50	5	5	25
15	15	5	5	25
	25	5	5	25
	50	5	5	25
Total # of Problem Instances				540

Besides, the processing times of jobs on each stage are parameters of our study, and in the computational study, we generate them randomly by using the uniform distribution between 0 and 100. The critical point here is that when analyzing different shop designs for some instances (the same number of jobs, the same number of operators, and the same number of experiments), the total processing time of a job must be the same for each shop design. For instance, in the case of 4 operators and 15 jobs, the processing time of job 1 can be as shown in Table 5.3. The distribution of processing times at the stages is so varied that the total processing time remains the same. Thus, we can compare all shop designs for a particular set of jobs and determine the best one.

Table 5.3 Example for Processing Time Distribution Based on Shop Design

Processing Time of J1	**Shop Design 2-2**	**Shop Design 2-1-1**	**Shop Design 1-2-1**	**Shop Design 1-1-2**	**Shop Design 1-1-1-1**
Stage 1	167	167	80	80	80
Buffer 1	0	0	0	0	0
Stage 2	73	40	127	87	87
Buffer 2	-	0	0	0	0
Stage 3	-	33	33	73	40
Buffer 3	-	-	-	-	0
Stage 4	-	-	-	-	33
Total Processing Time	240	240	240	240	240

On the other hand, for the MILP in the experiments, we use a time limit of 1 hour (3600 seconds), since the runs of MILP take too much time and we want to reach the solution in a reasonable time. If we do not use the time limit, we would probably reach a better solution. However, we waive better solutions in some cases to obtain the result in an acceptable time. If MILP finds the optimum solution in the time limit, we call this solution as an optimum solution with 0.00 % Gap. However, if MILP cannot find the optimum solution in one hour, we call the solution, which is the best result of MILP found within the time limit, as the best solution with a Gap value different from 0.00 %.

Our proposed heuristic approaches are coded in C# programming language with the .NET framework, and the experiments are executed on a computer that consists of Intel Core i7-4510U CPU @ 2.00 GHz and 16 GB RAM. Likewise, MILP runs are executed with IBM ILOG CPLEX Optimization Studio software on the same computer.

5.1. Computational Results

In this section, we share and analyze the computational results of experiments for the number of jobs, the number of operators, and the shop design alternatives. First, we compare the proposed mathematical models and decide the more effective one, which gives better results according to experiments. Then we analyze the proposed heuristic methods by comparing them with the better mathematical model.

5.1.1. Comparison of the MILP Models

We execute all experiments by both mathematical models and indicate the results in this section. At the end of the comparison, we select the model which obtains preferable solutions. When we have a look at the average gap values in Table 5.4, we can clearly see that parallel flow shop based MILP provides better solutions for every number of operator and job value.

Table 5.4 Gap Values of the MILP Models

# of Operators	**# of Jobs**	**Shop Design**	**Gap Values (%)**					
			FFS Based MILP			**PFS Based MILP**		
			Avg	**Min**	**Max**	**Avg**	**Min**	**Max**
4	15	2-2	0.33	0.16	0.46	0.24	0.03	0.40
		2-1-1	2.82	0.00	7.13	0.30	0.00	1.44
		1-2-1	3.77	0.00	7.77	0.85	0.00	2.10
		1-1-2	3.15	0.00	4.30	0.34	0.00	1.04
		1-1-1-1	5.39	2.34	9.82	0.48	0.21	0.92
	25	2-2	0.50	0.05	1.59	0.37	0.02	1.05
		2-1-1	2.40	0.00	5.27	1.02	0.00	2.52
		1-2-1	4.65	2.26	5.61	3.61	1.24	4.84
		1-1-2	3.46	0.10	7.22	1.84	0.00	4.92
		1-1-1-1	3.58	0.72	7.71	2.98	0.72	6.14
	50	2-2	0.77	0.06	2.61	0.18	0.00	0.70
		2-1-1	1.19	0.66	2.76	0.27	0.00	0.95
		1-2-1	2.64	0.05	5.59	1.17	0.00	2.07
		1-1-2	3.21	0.00	6.73	1.70	0.00	2.63

Table 5.4 (continued)

		1-1-1-1	3.08	1.16	5.80	1.46	0.40	2.07
5	15	2-2-1	4.48	0.40	6.76	0.54	0.07	1.94
		2-1-2	2.24	0.00	4.84	0.69	0.00	1.53
		1-2-2	2.46	0.00	4.24	1.39	0.00	2.89
		2-1-1-1	5.52	0.00	9.06	1.84	0.00	4.64
		1-2-1-1	7.62	5.37	9.36	3.72	0.66	6.47
		1-1-2-1	5.04	2.94	10.36	1.93	0.11	3.60
		1-1-1-2	3.04	0.00	7.59	1.73	0.00	4.62
		1-1-1-1-1	7.81	5.22	12.17	3.37	1.11	4.48
	25	2-2-1	3.59	1.41	5.17	3.07	1.41	4.09
		2-1-2	2.70	1.57	3.88	2.39	0.79	3.75
		1-2-2	3.26	0.52	4.39	2.50	0.45	3.74
		2-1-1-1	4.43	1.54	7.30	3.10	1.49	4.90
		1-2-1-1	5.63	0.68	8.90	5.24	0.68	7.37
		1-1-2-1	6.05	3.58	8.57	5.11	2.40	7.03
		1-1-1-2	4.30	2.23	5.80	3.96	2.23	5.25
		1-1-1-1-1	6.31	2.52	8.66	4.99	2.52	7.89
	50	2-2-1	2.39	1.53	4.78	1.47	0.31	2.91
		2-1-2	3.58	1.83	4.41	1.68	0.00	2.92
		1-2-2	3.64	1.36	5.91	1.86	0.04	3.34
		2-1-1-1	5.03	1.88	8.15	1.87	0.63	2.98
		1-2-1-1	3.91	1.38	6.40	2.21	0.31	3.28
		1-1-2-1	5.84	1.53	10.50	2.50	1.36	3.52
		1-1-1-2	3.88	1.64	7.93	1.94	0.00	3.59
		1-1-1-1-1	4.09	1.39	8.00	2.34	0.70	3.78
6	15	2-2-2	7.12	4.82	8.57	3.92	0.51	5.57
		2-2-1-1	7.29	2.42	11.43	2.80	0.65	4.22
		2-1-2-1	7.03	4.96	9.69	2.32	0.00	4.56
		2-1-1-2	6.72	0.66	15.12	4.80	0.66	8.38
		1-2-2-1	9.79	8.43	12.47	2.18	0.19	8.33
		1-2-1-2	11.10	5.61	15.02	5.88	3.76	7.65
		1-1-2-2	10.16	8.19	13.85	6.12	3.95	8.33
		2-1-1-1-1	9.18	6.71	12.58	3.06	0.45	4.87
		1-2-1-1-1	10.56	6.49	13.19	2.71	0.20	6.73
		1-1-2-1-1	8.86	2.23	13.73	2.55	0.06	7.24

Table 5.4 (continued)

		1-1-1-2-1	9.05	1.52	13.20	4.30	0.16	8.08
		1-1-1-1-2	7.45	0.66	13.09	7.24	4.82	12.92
		1-1-1-1-1-1	11.50	4.36	15.00	4.25	0.20	8.63
	25	2-2-2	3.07	0.93	5.66	2.18	0.43	3.54
		2-2-1-1	5.16	3.24	7.33	4.23	1.58	6.63
		2-1-2-1	5.38	2.47	10.25	3.84	0.43	7.77
		2-1-1-2	3.33	0.00	6.93	2.38	0.00	4.08
		1-2-2-1	4.75	3.42	7.65	4.32	2.96	7.40
		1-2-1-2	5.72	3.94	7.86	3.72	2.06	5.64
		1-1-2-2	4.31	2.27	7.02	3.65	2.34	5.14
		2-1-1-1-1	5.29	3.41	6.80	3.87	0.43	6.14
		1-2-1-1-1	5.09	2.93	9.05	5.15	2.93	8.80
		1-1-2-1-1	7.90	6.65	9.19	6.49	3.94	9.73
		1-1-1-2-1	5.27	1.60	11.77	4.67	1.60	9.91
		1-1-1-1-2	4.94	1.17	8.95	3.65	1.17	4.82
		1-1-1-1-1-1	5.77	1.60	7.77	5.14	1.60	6.90
	50	2-2-2	3.14	1.12	4.44	1.88	0.49	3.25
		2-2-1-1	6.65	2.98	9.22	3.22	1.91	4.44
		2-1-2-1	5.80	2.55	10.87	2.22	1.33	3.07
		2-1-1-2	4.45	1.65	8.64	2.65	1.83	3.92
		1-2-2-1	5.77	2.79	9.11	3.90	2.78	5.36
		1-2-1-2	4.67	3.36	6.29	4.35	2.89	6.41
		1-1-2-2	5.39	3.10	9.56	4.17	2.89	6.23
		2-1-1-1-1	7.35	4.43	9.37	2.80	1.78	4.39
		1-2-1-1-1	6.68	4.26	8.24	3.83	2.26	5.16
		1-1-2-1-1	7.68	5.36	13.02	4.52	2.54	7.18
		1-1-1-2-1	7.03	5.14	8.83	4.03	2.72	5.52
		1-1-1-1-2	7.69	3.66	12.677	3.91	1.87	5.25
		1-1-1-1-1-1	6.26	3.50	8.69	4.21	2.08	6.14
10	15	2-2-2-2-2	14.31	7.46	17.90	7.16	6.48	7.53
		2-1-2-1-2-2	12.10	7.41	18.29	9.54	4.01	12.49
		1-1-2-1-2-2-1	11.29	5.64	18.38	5. 29	1.22	9.39
		1-1-2-1-1-2-1-1	7.98	2.01	20.68	5.78	3.01	10.72
		1-1-1-1-1-1-1-1-1-1	9.40	2.63	19.50	9.86	7.38	13.11
	25	2-2-2-2-2	12.32	9.08	14.52	11.23	8.53	13.56

Table 5.4 (continued)

		2-1-2-1-2-2	12.72	8.68	17.77	11.65	7.64	14.72
		1-1-2-1-2-2-1	17.40	13.37	19.85	14.75	10.20	17.46
		1-1-2-1-1-2-1-1	18.64	13.15	22.86	15.65	10.47	19.39
		1-1-1-1-1-1-1-1-1-1	17.42	10.90	24.43	14.84	7.00	19.75
	50	2-2-2-2-2	9.89	8.32	11.69	6.43	5.09	8.69
		2-1-2-1-2-2	12.83	9.97	17.37	7.45	3.83	11.16
		1-1-2-1-2-2-1	15.91	12.50	19.31	9.63	6.21	12.35
		1-1-2-1-1-2-1-1	15.43	9.45	19.35	9.07	6.46	10.49
		1-1-1-1-1-1-1-1-1-1	16.74	12.58	19.93	10.48	8.03	12.03
15	15	2-2-2-2-2-2-2-1	18.93	14.88	23.34	10.68	0.75	14.80
		2-2-1-1-2-1-1-2-1-1-1	16.18	12.52	18.03	14.56	7.90	20.69
		1-1-1-1-2-2-2-2-1-2	16.97	9.68	21.80	15.49	12.74	18.74
		1-2-2-1-1-1-1-1-1-2-2	16.75	9.62	20.29	13.21	9.68	17.59
		1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	12.27	3.94	19.06	18.60	16.60	21.26
	25	2-2-2-2-2-2-2-1	26.54	23.50	29.39	24.04	22.04	26.95
		2-2-1-1-2-1-1-2-1-1-1	31.46	25.66	34.29	26.74	23.58	29.09
		1-1-1-1-2-2-2-2-1-2	29.48	27.90	31.31	26.06	21.82	28.89
		1-2-2-1-1-1-1-1-1-2-2	30.04	26.11	34.24	27.07	24.13	28.12
		1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	34.69	31.17	38.76	28.87	25.33	31.93
	50	2-2-2-2-2-2-2-1	22.06	17.90	24.61	18.29	14.14	20.39
		2-2-1-1-2-1-1-2-1-1-1	27.28	24.42	30.50	19.65	16.70	23.18
		1-1-1-1-2-2-2-2-1-2	25.55	20.88	28.63	18.53	15.06	21.72
		1-2-2-1-1-1-1-1-1-2-2	25.18	21.78	28.54	16.69	12.43	20.19
		1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	28.10	25.33	32.76	22.34	20.09	25.41

Furthermore, another significant point is how many times the MILP has reached the optimal solution. As seen in Table 5.5, FFS based MILP model obtains the optimal solution for 11 times out of 540 with a ratio of 2.0%, and PFS based MILP model reaches 52 optimal solutions with a ratio of 9.6%. This table shows that PFS based MILP model is superior to FFS based model. Hence, we analyze and evaluate the proposed heuristics by comparing them against PFS based MILP in the next section.

Table 5.5 Frequency of Optimal Solutions for Each MILP Model

# of Operators	**# of Jobs**	**Shop Design**	**# of Exp.**	**# of Optimal Solutions**	
				FFS Based MILP	PFS Based MILP
4	15	2-1-1	5	2	2
		1-2-1	5	1	5
		1-1-2	5	1	5
		1-1-1-1	5	0	3
	25	2-1-1	5	1	2
		1-1-2	5	0	2
	50	2-2	5	0	1
		2-1-1	5	0	3
		1-2-1	5	0	1
		1-1-2	5	1	1
5	15	2-2-1	5	0	2
		2-1-2	5	1	3
		1-2-2	5	1	2
		2-1-1-1	5	1	2
		1-2-1-1	5	0	2
		1-1-1-2	5	1	2
		1-1-1-1-1	5	0	1
	50	2-1-2	5	0	1
		1-1-1-2	5	0	1
6	15	2-2-1-1	5	0	2
		2-1-2-1	5	0	2
		1-2-2-1	5	0	1
		1-2-1-1-1	5	0	1
		1-1-2-1-1	5	0	1
		1-1-1-2-1	5	0	2
		1-1-1-1-1-1	5	0	1
	25	2-1-1-2	5	1	1
		Total	540	11	52
			Ratio	2.0%	9.6%

5.1.2. Comparison of the Heuristic Approaches

In this section, we consider PFS based MILP, SPT&NEH, LPT&NEH, SPT, LPT, and "Best of Heuristics" while analyzing the computational results. Therefore, when the term MILP is used later in this section, PFS based MILP model need to be understood.

We evaluate the performance of the solution methods regarding the following measures:

- Average gap value: The gap value of each experiment for each solution approach is calculated, and then their average values are calculated
- Total number of the best solutions found: 540 problem instances are analyzed, and the frequencies of the best solutions are observed, that is, how many times each solution approach gives the best solution.
- Average solution time: The solution time of each experiment for each solution approach is calculated, and then their average values are calculated.
- Best shop design: "how many times each shop design alternative provides the best solution" is also analyzed.

When performing the analysis above, we categorize the experiments by the number of operators and the number of jobs to obtain a more precise result.

The gap values of all solution methods for each experiment are indicated in Table 5.6. As shown in the table, heuristic methods provide pretty good solutions. For example, when we look at the experiments with 4-operator and 2-2 shop design, the best of heuristics gives us the result with 2.21% average gap value for 15 jobs, 1.15% gap value for 25 jobs, and 0.45% gap value for 50 jobs. Likewise, according to the results of experiments with 5-operator and 2-2-1 shop design, the best of heuristics reaches the solution with 2.45% average gap value for 15 jobs, 1.61% average gap value for 25 jobs, and 0.48% average gap value for 50 jobs. We obtain satisfactory results with 6-operator, 10-operator, and 15-operator as well.

Besides, changes in the average gap values of the MILP and “Best of Heuristics” experiments depending on the number of jobs are indicated in Figures 5.1 and 5.2. The average gap value of “Best of Heuristics” experiments tends to decrease with the increase in the workforce level, that is the number of operators. However, it is not the same for the MILP results. The average gap value of 25-operator is higher than both 15-operator and 50-operator values, which means that the gap value rises first and then declines.

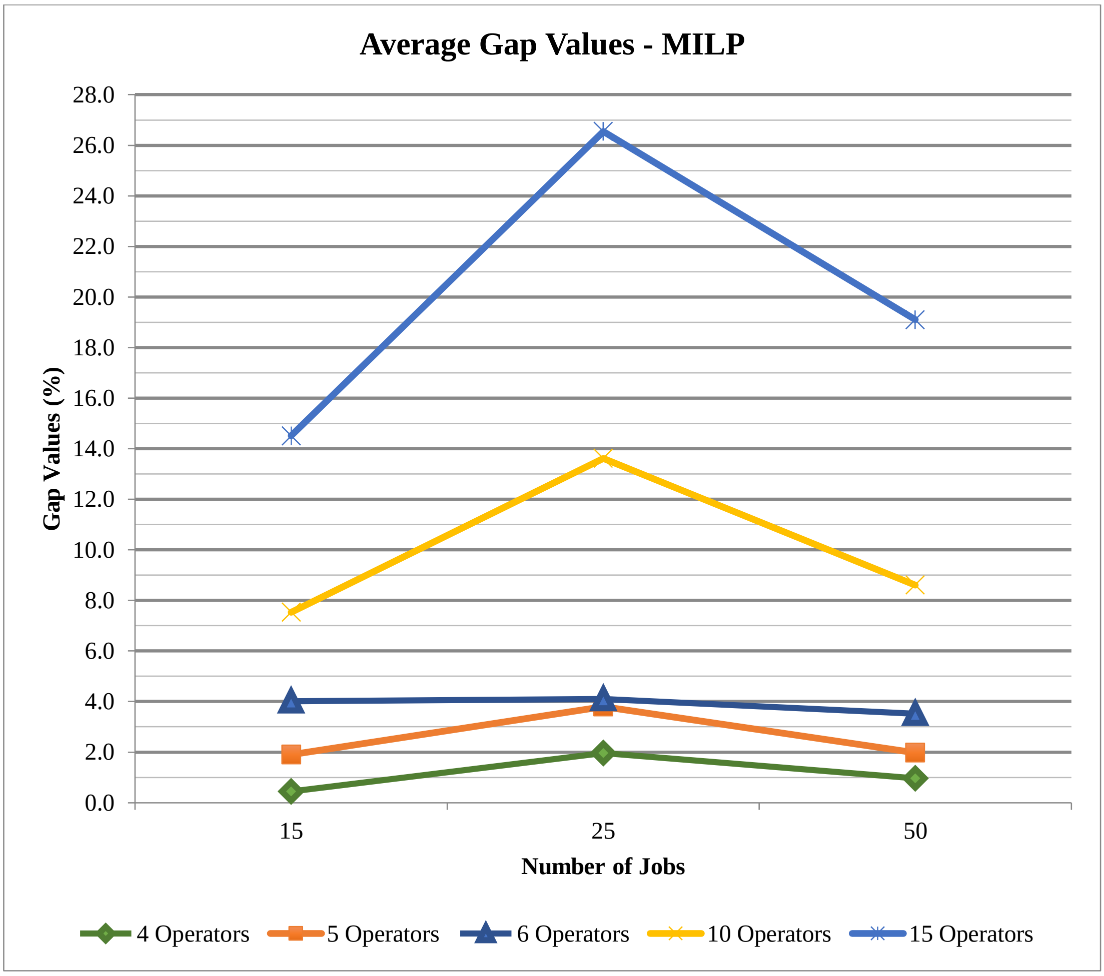

Figure 5.1 Change in the Average Gap Values of the MILP Experiments Based on the Number of Jobs

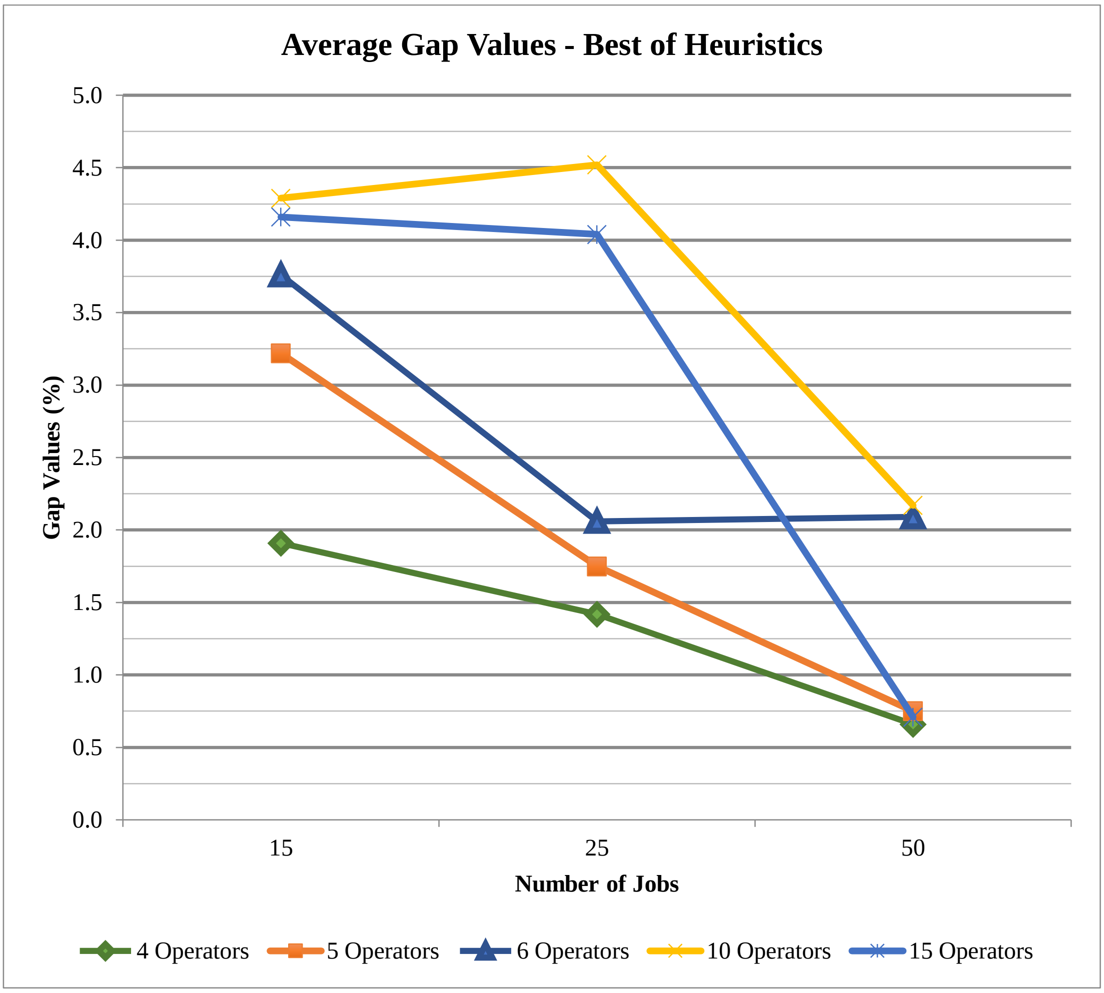

Figure 5.2 Change in the Average Gap Values of the "Best of Heuristics" Experiments Based on the Number of Jobs

Table 5.6 Gap Values of the Solution Approaches

# of Opr.	# of Jobs	Shop Design	Gap Values (%)																	
			MILP			SPT&NEH			LPT&NEH			SPT			LPT			Best of Heuristics		
			Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
4	15	2-2	0.24	0.03	0.40	5.63	3.55	12.07	2.21	0.52	3.52	13.18	7.16	19.11	17.80	8.65	24.60	2.21	0.52	3.52
		2-1-1	0.30	0.00	1.44	4.68	0.00	6.76	0.75	0.00	1.45	12.35	7.57	17.22	18.15	7.33	27.23	0.75	0.00	1.45
		1-2-1	0.85	0.00	2.10	2.49	0.00	4.13	1.36	0.00	3.95	15.88	8.67	26.45	18.35	11.74	26.38	1.36	0.00	3.95
		1-1-2	0.34	0.00	1.04	5.44	2.02	9.12	4.86	0.00	10.36	12.11	6.99	18.80	15.90	12.05	18.31	4.29	0.00	9.12
		1-1-1-1	0.48	0.21	0.92	1.41	0.00	3.32	1.15	0.00	2.71	13.25	10.87	16.36	16.82	11.08	24.85	0.93	0.00	2.71
	25	2-2	0.37	0.02	1.05	3.96	0.66	5.61	1.15	0.42	2.23	11.69	5.51	15.44	9.01	4.67	13.30	1.15	0.42	2.23
		2-1-1	1.02	0.00	2.52	2.29	0.00	6.58	1.00	0.00	2.91	10.15	7.15	13.03	10.99	7.94	14.24	1.00	0.00	2.91
		1-2-1	3.61	1.24	4.84	3.26	0.14	5.15	2.18	0.00	5.06	14.83	10.18	20.59	13.08	7.04	19.62	1.91	0.00	3.69
		1-1-2	1.84	0.00	4.92	3.17	0.10	5.07	2.65	0.00	6.06	14.18	7.64	18.92	8.73	6.56	10.67	2.42	0.00	5.07
		1-1-1-1	2.98	0.72	6.14	2.12	0.00	6.53	0.77	0.00	1.96	13.81	7.64	22.17	13.19	9.87	17.41	0.62	0.00	1.96
	50	2-2	0.18	0.00	0.70	1.14	0.08	2.09	0.68	0.20	1.00	7.08	1.14	10.45	6.20	4.34	8.77	0.45	0.08	0.87
		2-1-1	0.27	0.00	0.95	0.60	0.00	1.68	0.08	0.00	0.40	6.36	2.58	12.14	8.50	5.06	12.27	0.08	0.00	0.40
		1-2-1	1.17	0.00	2.07	1.61	0.00	3.69	0.93	0.39	1.85	11.17	5.40	19.50	10.93	5.22	14.78	0.75	0.00	1.85
		1-1-2	1.70	0.00	2.63	1.31	0.54	1.82	1.05	0.20	2.71	14.58	2.49	21.29	11.72	7.43	17.48	0.72	0.20	1.44
		1-1-1-1	1.46	0.40	2.07	1.43	0.06	4.46	1.96	0.00	4.10	13.70	4.22	20.75	12.42	6.17	20.00	1.32	0.00	3.98
5	15	2-2-1	0.54	0.07	1.94	5.05	0.20	7.31	2.75	1.71	3.81	19.60	10.50	28.87	19.30	15.30	24.80	2.45	0.20	3.81
		2-1-2	0.69	0.00	1.53	5.52	4.18	6.78	3.51	2.37	5.41	22.27	17.74	30.86	24.34	19.23	28.59	3.51	2.37	5.41
		1-2-2	1.39	0.00	2.89	7.46	0.22	12.54	3.82	0.00	8.89	25.21	22.17	31.75	19.44	7.39	29.83	3.71	0.00	8.29
		2-1-1-1	1.84	0.00	4.64	3.00	0.00	6.39	2.38	0.00	5.44	14.81	0.00	22.01	18.88	12.40	29.62	2.19	0.00	5.44
		1-2-1-1	3.72	0.66	6.47	5.56	-0.52	16.67	3.50	1.56	8.27	17.50	0.00	27.43	18.77	10.05	23.88	3.09	-0.52	8.27

Table 5.6 (continued)

		1-1-2-1	1.93	0.11	3.60	3.98	0.10	6.71	6.18	1.61	11.06	23.47	21.63	25.65	19.00	9.83	29.21	3.73	0.10	6.71
		1-1-1-2	1.73	0.00	4.62	6.42	4.28	10.80	4.21	0.52	12.54	25.57	21.82	33.44	19.37	12.17	24.28	2.80	0.52	5.51
		1-1-1-1-1	3.37	1.11	4.48	5.32	0.00	8.34	5.21	1.39	11.52	21.08	0.00	32.49	20.93	10.05	25.69	4.27	0.00	8.18
	25	2-2-1	3.07	1.41	4.09	2.84	0.00	7.14	1.61	0.00	5.06	14.09	4.55	24.26	15.86	12.26	20.91	1.61	0.00	5.06
		2-1-2	2.39	0.79	3.75	5.13	2.75	8.70	1.97	0.36	5.42	18.80	17.18	22.00	21.92	18.99	26.47	1.43	0.36	2.75
		1-2-2	2.50	0.45	3.74	3.72	0.00	6.29	1.64	-0.23	3.39	20.14	12.07	31.24	17.92	15.55	23.31	1.44	-0.23	3.39
		2-1-1-1	3.10	1.49	4.90	4.11	0.00	9.86	1.79	0.49	4.12	13.63	7.06	18.76	18.08	13.19	23.88	1.67	0.00	4.12
		1-2-1-1	5.24	0.68	7.37	3.36	0.00	6.64	1.73	0.00	3.47	17.60	6.88	27.21	18.71	14.25	24.58	1.63	0.00	2.99
		1-1-2-1	5.11	2.40	7.03	4.66	0.00	11.79	2.86	0.55	6.30	16.93	6.63	26.52	16.84	14.25	19.50	2.38	0.00	6.30
		1-1-1-2	3.96	2.23	5.25	3.40	0.41	6.28	3.12	0.33	5.93	22.26	16.89	30.61	19.95	17.48	24.56	2.38	0.33	5.93
		1-1-1-1-1	4.99	2.52	7.89	3.89	0.18	8.91	2.39	0.33	4.55	18.09	9.46	28.35	19.70	14.25	24.27	1.45	0.18	3.47
	50	2-2-1	1.47	0.31	2.91	1.24	0.00	3.94	0.48	-0.04	1.78	10.18	4.05	15.48	10.59	3.49	19.58	0.48	-0.04	1.78
		2-1-2	1.68	0.00	2.92	3.42	1.80	5.39	1.25	0.36	2.73	17.46	10.40	29.32	15.24	0.72	29.32	1.25	0.36	2.73
		1-2-2	1.86	0.04	3.34	1.46	0.62	2.22	0.50	0.25	0.70	14.42	12.73	16.10	15.23	13.37	17.68	0.48	0.25	0.67
		2-1-1-1	1.87	0.63	2.98	2.34	-1.07	5.50	0.23	-1.07	0.80	11.87	5.47	19.37	14.82	4.47	34.26	0.23	-1.07	0.80
		1-2-1-1	2.21	0.31	3.28	0.83	-1.07	2.52	1.23	0.00	3.04	12.91	4.66	18.59	16.01	4.07	25.47	0.45	-1.07	2.52
		1-1-2-1	2.50	1.36	3.52	2.40	-1.07	7.08	1.34	-1.07	3.19	13.39	4.61	18.59	14.68	4.77	24.00	1.34	-1.07	3.19
		1-1-1-2	1.94	0.00	3.59	2.03	0.22	4.62	0.82	0.03	2.06	15.84	11.03	20.46	16.62	7.64	22.84	0.60	0.03	1.22
		1-1-1-1-1	2.34	0.70	3.78	2.09	-0.24	6.15	1.25	-0.24	4.63	13.67	6.11	20.21	17.91	5.95	35.39	1.18	-0.24	4.63
6	15	2-2-2	3.92	0.51	5.57	9.90	6.13	14.16	3.70	1.02	6.73	25.99	20.66	35.51	31.68	13.79	49.59	3.70	1.02	6.73
		2-2-1-1	2.80	0.65	4.22	3.08	0.00	9.38	1.21	0.00	2.51	16.66	9.65	29.77	21.24	11.25	27.39	0.79	0.00	2.51
		2-1-2-1	2.32	0.00	4.56	1.61	0.41	2.62	1.27	0.82	1.87	14.56	5.83	20.51	20.22	11.22	30.03	1.19	0.41	1.87
		2-1-1-2	4.80	0.66	8.38	7.76	3.95	10.45	6.55	4.31	9.26	23.44	12.24	33.88	29.18	9.06	39.71	6.09	3.95	9.26
		1-2-2-1	2.18	0.19	8.33	4.51	0.00	9.42	2.37	0.00	5.77	19.18	9.93	28.59	23.72	16.34	33.04	1.75	0.00	5.77
		1-2-1-2	5.88	3.76	7.65	6.90	5.88	8.53	5.63	3.43	7.96	25.13	17.33	29.69	28.21	19.70	39.11	5.27	3.43	7.51
		1-1-2-2	6.12	3.95	8.33	10.44	7.15	11.92	7.84	4.15	12.64	30.78	25.85	37.62	29.38	19.28	39.05	7.69	4.15	11.92

Table 5.6 (continued)

		2-1-1-1-1	3.06	0.45	4.87	5.32	1.59	9.26	3.75	0.85	7.32	20.09	7.76	32.94	23.05	12.13	31.32	2.89	0.85	7.32
		1-2-1-1-1	2.71	0.20	6.73	6.85	3.81	11.05	4.51	1.06	6.89	21.40	12.13	28.70	21.48	17.63	24.85	3.27	1.06	4.50
		1-1-2-1-1	2.55	0.06	7.24	4.99	1.51	8.01	3.84	0.62	5.65	22.25	18.54	27.48	24.84	18.60	31.00	3.41	0.62	5.65
		1-1-1-2-1	4.30	0.16	8.08	4.00	0.00	9.10	3.91	1.06	7.81	22.42	16.21	33.51	25.58	16.83	34.08	3.12	0.00	7.81
		1-1-1-1-2	7.24	4.82	12.92	6.92	4.47	10.09	8.14	3.95	14.54	30.07	23.70	36.51	31.17	18.59	44.63	6.17	3.95	8.00
		1-1-1-1-1-1	4.25	0.20	8.63	5.83	1.51	9.67	4.59	1.72	6.58	24.08	17.64	31.48	27.37	17.36	34.58	3.54	1.51	6.58
		2-2-2	2.18	0.43	3.54	3.84	-1.31	9.45	0.78	-1.37	2.62	20.82	11.89	30.82	19.68	13.46	24.80	0.78	-1.37	2.62
		2-2-1-1	4.23	1.58	6.63	4.65	2.09	8.18	1.81	0.00	6.08	18.09	9.97	26.94	18.15	7.66	24.95	1.49	0.00	4.51
		2-1-2-1	3.84	0.43	7.77	3.50	1.28	5.94	1.87	0.34	3.33	16.95	10.53	25.11	19.00	13.30	23.63	1.87	0.34	3.33
		2-1-1-2	2.38	0.00	4.08	5.10	1.11	8.70	1.95	0.00	5.19	21.66	14.27	28.05	27.80	20.89	35.49	1.95	0.00	5.19
		1-2-2-1	4.32	2.96	7.40	4.58	1.25	8.34	3.35	0.56	5.26	21.36	6.52	32.92	18.65	11.59	28.10	2.93	0.56	5.26
		1-2-1-2	3.72	2.06	5.64	5.25	3.28	6.68	4.55	1.78	7.69	23.22	16.09	29.93	22.43	13.33	29.40	4.11	1.78	6.68
	25	1-1-2-2	3.65	2.34	5.14	5.81	3.97	11.83	4.11	1.28	6.11	25.02	13.71	31.18	20.29	13.55	29.72	3.88	1.28	5.90
		2-1-1-1-1	3.87	0.43	6.14	2.81	-0.38	5.44	1.77	0.47	3.57	18.95	8.28	26.43	18.21	8.73	24.46	0.98	-0.38	2.72
		1-2-1-1-1	5.15	2.93	8.80	3.41	0.78	7.04	2.44	-0.31	4.50	20.36	9.53	27.30	21.40	10.71	32.16	2.37	-0.31	4.50
		1-1-2-1-1	6.49	3.94	9.73	3.68	-0.25	7.89	3.50	1.43	4.60	21.67	8.34	26.99	18.73	13.39	24.05	1.95	-0.25	3.54
		1-1-1-2-1	4.67	1.60	9.91	3.79	-0.94	5.73	2.29	0.12	4.89	23.77	8.84	32.20	21.25	18.11	26.28	1.29	-0.94	4.89
		1-1-1-1-2	3.65	1.17	4.82	2.77	0.80	6.11	1.89	-0.06	4.32	24.13	12.36	29.02	24.39	21.36	26.88	1.15	-0.06	3.38
		1-1-1-1-1-1	5.14	1.60	6.90	4.59	2.32	6.15	2.58	0.37	5.82	22.87	11.54	29.02	26.21	20.98	30.34	2.06	0.37	4.95
		2-2-2	1.88	0.49	3.25	2.69	0.41	7.14	1.58	-0.48	4.55	17.07	12.88	21.26	14.77	10.53	18.83	1.58	-0.48	4.55
		2-2-1-1	3.22	1.91	4.44	1.87	0.42	3.31	1.42	0.24	3.88	18.11	14.45	22.66	14.85	10.86	17.85	1.31	0.24	3.31
		2-1-2-1	2.22	1.33	3.07	3.49	0.10	6.61	2.91	-0. 58	6.06	17.18	11.12	23.90	13.74	8.18	17.99	2.81	-0.58	6.06
	50	2-1-1-2	2.65	1.83	3.92	4.01	1.43	8.50	1.65	-0.21	5.50	22.03	14.53	28.30	16.34	9.46	27.06	1.37	-0.21	4.11
		1-2-2-1	3.90	2.78	5.36	2.11	0.00	4.15	1.49	0.00	4.52	18.18	11.11	27.05	12.40	8.64	15.06	1.42	0.00	4.15
		1-2-1-2	4.35	2.89	6.41	1.98	0.84	3.51	1.61	0.72	2.60	17.65	13.39	24.61	13.69	11.21	17.19	1.07	0.72	2.13
		1-1-2-2	4.17	2.89	6.23	2.65	0.60	4.51	0.99	-0.93	2.38	19.99	13.15	36.09	12.93	11.43	14.30	0.72	-0.93	2.38

Table 5.6 (continued)

		2-1-1-1-1	2.80	1.78	4.39	4.45	2.29	6.98	3.21	0.00	7.24	24.09	20.11	33.25	18.58	9.68	25.68	2.93	0.00	6.98
		1-2-1-1-1	3.83	2.26	5.16	4.58	2.19	6.80	4.82	2.18	10.06	22.66	17.51	32.18	16.00	12.54	20.77	3.25	2.18	5.10
		1-1-2-1-1	4.52	2.54	7.18	2.53	0.80	4.59	3.50	0.93	6.57	21.73	18.28	29.57	16.05	12.59	20.13	2.15	0.80	4.59
		1-1-1-2-1	4.03	2.72	5.52	3.64	1.73	5.86	2.93	0.93	3.65	21.54	15.29	26.76	16.30	9.29	23.34	2.77	0.93	3.54
		1-1-1-1-2	3.91	1.87	5.25	3.58	1.73	5.17	2.54	0.77	3.62	20.69	17.00	25.81	15.80	9.12	22.31	2.54	0.77	3.62
		1-1-1-1-1-1	4.21	2.08	6.14	3.52	2.06	5.62	4.46	0.93	7.60	24.91	22.94	30.66	18.35	12.26	24.81	3.30	0.93	5.62
10	15	2-2-2-2-2	7.16	6.48	7.53	8.14	6.20	12.19	4.00	2.55	5.30	28.28	20.09	35.16	25.13	14.75	33.30	4.00	2.55	5.30
		2-1-2-1-2-2	9.54	4.01	12.49	6.46	3.46	9.49	3.32	2.06	4.52	27.47	19.22	34.81	25.26	17.74	32.46	3.32	2.06	4.52
		1-1-2-1-2-2-1	5.29	1.22	9.39	5.59	1.71	8.87	5.86	3.10	7.91	28.04	19.20	42.08	22.24	15.73	27.54	5.09	1.71	7.91
		1-1-2-1-1-2-1-1	5.78	3.01	10.72	4.99	2.42	7.04	6.48	3.60	9.95	26.52	18.18	38.92	22.62	15.54	32.25	4.63	2.42	7.03
		1-1-1-1-1-1-1-1-1-1	9.86	7.38	13.11	4.40	2.71	5.63	6.70	4.64	9.35	28.79	19.17	39.58	26.61	17.41	34.26	4.38	2.71	5.63
	25	2-2-2-2-2	11.23	8.53	13.56	7.28	5.16	9.25	2.72	1.43	5.22	27.49	20.51	37.31	24.71	17.78	29.49	2.71	1.43	5.16
		2-1-2-1-2-2	11.65	7.64	14.72	4.29	1.07	9.21	2.91	0.38	4.92	26.88	18.51	33.06	26.37	24.11	28.33	2.45	0.38	4.92
		1-1-2-1-2-2-1	14.75	10.20	17.46	7.69	3.48	12.64	5.88	3.24	7.95	26.42	16.70	33.06	28.06	17.66	35.97	4.95	3.24	7.95
		1-1-2-1-1-2-1-1	15.65	10.47	19.39	9.44	5.01	11.51	6.90	4.05	9.53	27.25	19.02	33.65	27.53	19.82	37.79	6.90	4.05	9.53
		1-1-1-1-1-1-1-1-1-1	14.84	7.00	19.75	5.92	2.92	8.83	7.14	4.65	11.00	28.28	20.31	32.27	27.02	20.15	34.20	5.57	2.92	8.34
	50	2-2-2-2-2	6.43	5.09	8.69	1.35	-0.10	2.56	0.29	-0. 45	1.23	18.47	12.44	27.51	17.41	13.02	20.40	0.18	-0.45	1.23
		2-1-2-1-2-2	7.45	3.83	11.16	3.67	-0.39	7.54	2.04	-2.50	5. 08	20.42	11.75	27.26	19.97	14.83	22.98	2.04	-2.50	5.08
		1-1-2-1-2-2-1	9.63	6.21	12.35	3.76	1.44	7.19	3.56	-1.54	6.66	21.09	12.50	26.11	20.08	16.82	23.55	2.59	-1.54	4.54
		1-1-2-1-1-2-1-1	9.07	6.46	10.49	4.46	1.07	9.26	4.12	-2.76	8.74	22.00	17.76	25.23	22.97	18.28	25.93	3.43	-2.76	8.74
		1-1-1-1-1-1-1-1-1-1	10.48	8.03	12.03	2.62	-0.74	5.32	5.44	3.96	8.03	22.10	16.93	25.74	24.15	21.23	26.42	2.62	-0.74	5.32
15	15	2-2-2-2-2-2-2-1	10.68	0.75	14.80	4.59	2.82	8.86	3.45	1.85	4.89	20.21	9.06	32.42	21.65	15.55	27.94	2.99	1.85	4.40

Table 5.6 (continued)

		2-2-1-1-2-1-1-2-1-1-1	14.56	7.90	20.69	3.73	1.68	6.98	5.42	1.85	8.02	19.75	13.91	30.94	19.31	17.04	22.44	3.05	1.68	5.13
		1-1-1-1-2-2-2-2-1-2	15.49	12.74	18.74	7.14	4.65	10.38	6.51	2.36	9.16	21.55	12.02	36.19	20.94	19.02	22.76	6.05	2.36	9.16
		1-2-2-1-1-1-1-1-1-2-2	13.21	9.68	17.59	8.12	4.39	10.19	4.47	-2.25	7.67	20.51	11.23	34.94	22.22	13.17	33.53	4.06	-2.25	7.13
		1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	18.60	16.60	21.26	5.27	3.54	8.45	5.55	3.07	8.45	22.20	13.59	35.26	21.74	16.56	26.62	4.67	3.07	8.45
	25	2-2-2-2-2-2-2-1	24.04	22.04	26.95	4.08	2.13	8.13	0.82	-1.01	2.07	18.58	13.58	22.84	19.48	17.81	21.30	0.82	-1.01	2.07
		2-2-1-1-2-1-1-2-1-1-1	26.74	23.58	29.09	5.67	4.50	7.34	4.05	2.25	5.13	17.76	10.70	21.89	21.57	18.16	25.19	3.96	2.25	5.13
		1-1-1-1-2-2-2-2-1-2	26.06	21.82	28.89	5.64	3.71	7.18	6.66	2.97	11.39	19.76	14.48	23.86	22.92	18.11	30.48	5.34	2.97	7.18
		1-2-2-1-1-1-1-1-1-2-2	27.07	24.13	28.12	6.94	3.58	10.64	6.37	2.85	8.46	19.86	16.51	22.07	21.77	11.75	35.05	5.31	2.85	7.70
		1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	28.87	25.33	31.93	5.06	3.92	7.42	5.40	4.11	8.43	24.13	19.45	31.14	21.74	15.61	24.65	4.75	3.92	7.42
	50	2-2-2-2-2-2-2-1	18.29	14.14	20.39	-1.35	-2.32	0.06	-3.25	-3.86	-1.81	14.55	12.24	17.17	14.80	12.89	17.38	-3.25	-3.86	-1.81
		2-2-1-1-2-1-1-2-1-1-1	19.65	16.70	23.18	3.65	-1.84	7.22	2.49	-3.20	6.18	18.49	13.29	24.32	20.71	15.15	25.30	2.49	-3.20	6.18
		1-1-1-1-2-2-2-2-1-2	18.53	15.06	21.72	2.82	-0.56	8.77	1.89	-3.12	8.77	18.36	9.89	35.75	19.89	8.79	30.81	1.89	-3.12	8.77
		1-2-2-1-1-1-1-1-1-2-2	16.69	12.43	20.19	3.71	2.36	6.66	3.80	0.67	8.76	20.15	16.26	25.59	21.45	14.40	28.57	2.23	0.67	2.85
		1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	22.34	20.09	25.41	0.88	-1.46	4.06	0.44	-3.50	4.78	18.04	13.01	21.12	19.61	9.84	26.09	0.18	-3.50	4.06

Furthermore, the average gap value related to the change in the number of operators is crucial to evaluate the performance of MILP and heuristic methods. As shown in Figures 5.3 and 5.4, while the value is continuously increasing with the increment of the number of operators in the MILP experiments, it increases until 10-operator and then starts to decrease in the "Best of Heuristics" experiments.

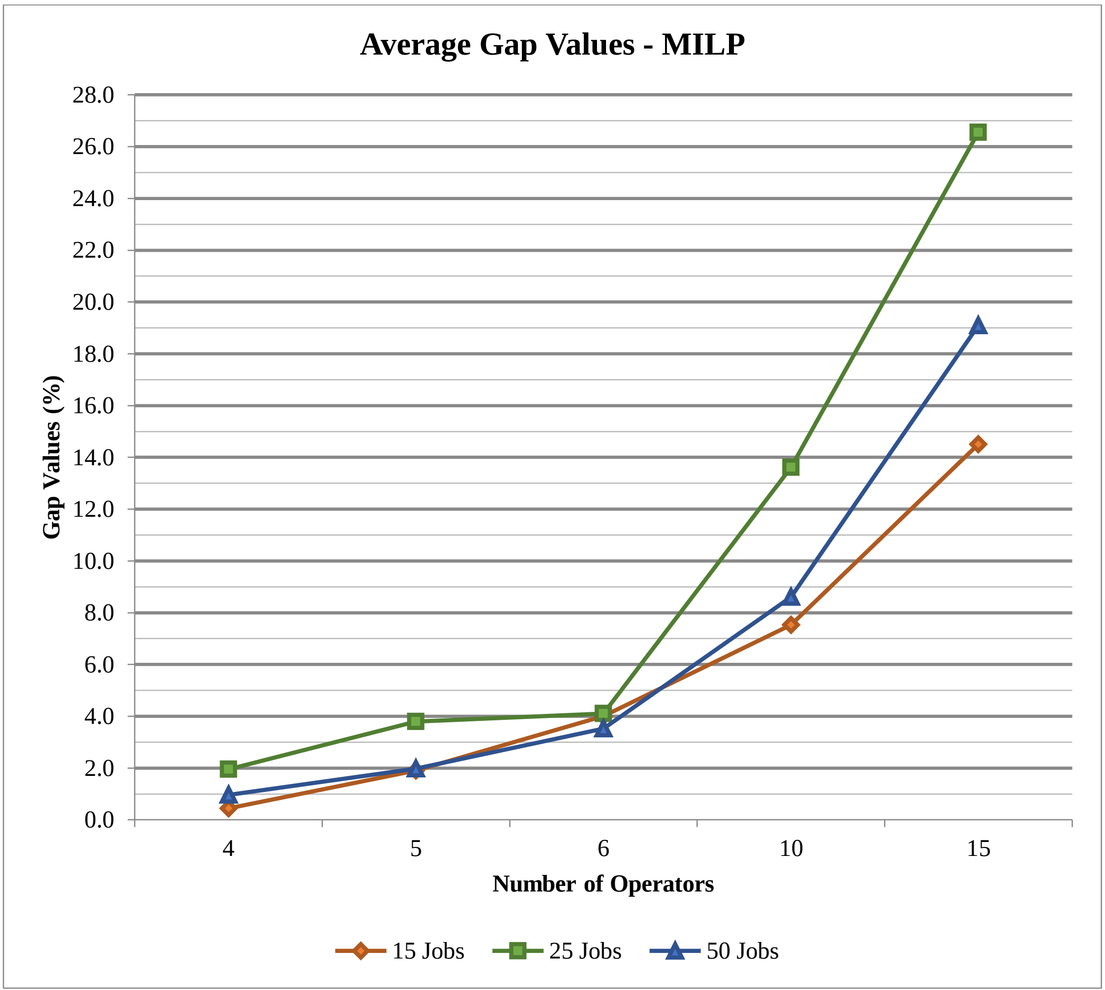

Figure 5.3 Change in the Average Gap Values by the MILP Model Based on the Number of Operators

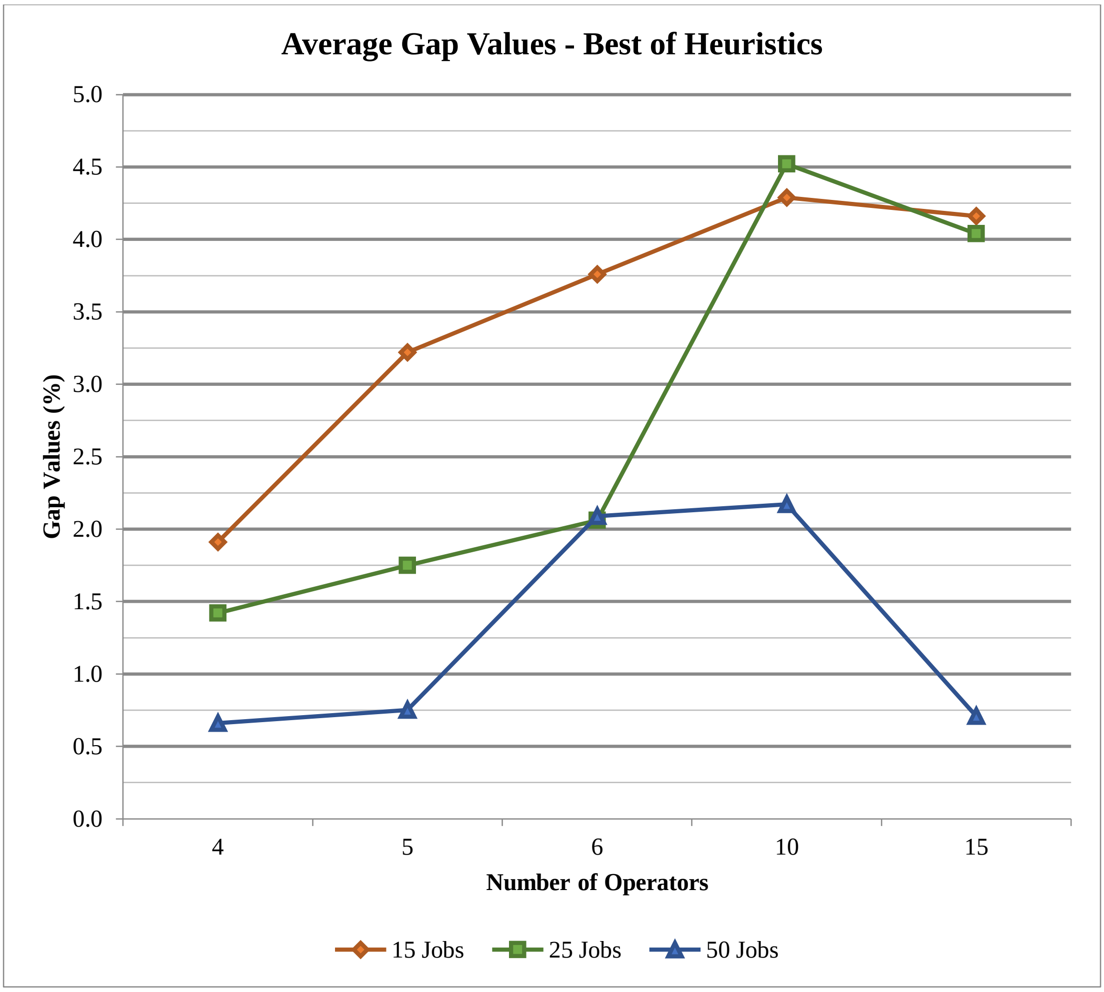

Figure 5.4 Change in the Average Gap Values by the "Best of Heuristics" Based on the Number of Operators

On the other hand, it is also vital for us to know "how many times the solution method has reached the best result". As can be seen from Table 5.7, while the MILP model finds the best solution in 505 out of 540 problem instances with 93.5%, the "best of heuristics" reaches the best solution in 79 problem instances with 14.6%. Among the heuristic methods, the LPT&NEH Algorithm stands out as the most promising method, achieving the best solution with a rate of 11.5%.

Table 5.7 The Frequency of the Best Solution for Each Solution Method

# of Operators	# of Jobs	# of Problem Instances	MILP	SPT&NEH Algorithm	LPT&NEH Algorithm	SPT	LPT	Best of Heuristic
4	15	25	25	4	7	0	0	8
	25	25	25	2	6	0	0	6
	50	25	25	3	5	0	0	6
5	15	40	39	3	3	2	0	5
	25	40	39	6	4	0	0	7
	50	40	35	6	7	0	0	9
6	15	65	65	5	2	0	0	6
	25	65	59	3	5	0	0	8
	50	65	60	1	7	0	0	7
10	15	25	25	0	0	0	0	0
	25	25	25	0	0	0	0	0
	50	25	19	1	5	0	0	6
15	15	25	24	0	1	0	0	1
	25	25	24	0	1	0	0	1
	50	25	16	0	9	0	0	9
Total		540	505	34	62	2	0	79
Ratio			93.5%	6.7%	11.5%	0.4%	0.0%	14.6%

Another significant issue for our study is the solution time of the heuristic approaches. If we look at the average solution times in Table 5.8, it is obvious that the heuristic methods are very effective and efficient compared to the MILP Model. In the experiments executed with 15 jobs, while the MILP model obtains the best solution in 2850 seconds on the average, the "best of heuristics" obtains in about 0.3 seconds. Likewise, in 25-job and 50-job experiments, the MILP model reaches the result in about 3475 and 3400 seconds respectively, and the "best of heuristics" comes up with a solution in the same experiments in about 1.9 and 24.9 seconds.

Table 5.8 Computational Times of the Solution Methods

# of Operators	# of Jobs	Average Solution Times (Seconds)					
		MILP	SPT&NEH	LPT&NEH	SPT	LPT	Best of Heuristic
4	15	1547.2	0.1	0.1	0.01	0.01	0.1
	25	3024.7	1.0	1.0	0.01	0.01	1.0
	50	2810.5	10.4	10.4	0.02	0.02	10.4
5	15	2385.5	0.2	0.2	0.01	0.01	0.2
	25	3600	1.2	1.2	0.01	0.01	1.2
	50	3426.1	14.9	14.9	0.02	0.02	14.9
6	15	3106.9	0.3	0.3	0.01	0.01	0.3
	25	3544.7	1.3	1.3	0.01	0.01	1.3
	50	3600	16.5	16.5	0.02	0.02	16.5
10	15	3600	0.5	0.5	0.01	0.01	0.5
	25	3600	2.0	2.0	0.01	0.01	2.0
	50	3600	24.1	24.1	0.04	0.04	24.1
15	15	3600	0.65	0.65	0.01	0.01	0.65
	25	3600	3.8	3.8	0.02	0.02	3.8
	50	3600	58.2	58.2	0.08	0.08	58.2

Another crucial part of our study consists of searching for the efficiency of the shop design alternatives. We execute the experiments with three alternatives of the number of jobs, 15, 25, and 50, for each number of operators. For each number of job option, we conduct five experiments, which means 15 experiments for each "number of operators". If we consider shop design alternatives for each operator level, the total number of problem instances reaches 540. The numbers of times in which the shop design alternative gives the best solution are summarized in Table 5.9. Certain shop design alternatives stand out among others with their performances, such as 2-2 and 2-2-2 alternatives. On the other hand, although the first shop designs in 10-operator and 15-operator experiments reach the best results more than the other designs, we cannot consider them as the best shop design alternatives because we do not analyze the entire shop design configurations for 10- and 15-operator cases.

Table 5.9 Performance of Shop Design Alternatives

4-Operator		**5-Operator**		**6-Operator**		**10-Operator**		**15-Operator**	
Shop Design	# of Best Solution	Shop Design	# of Best Solution	Shop Design	# of Best Solution	Shop Design	# of Best Solution	Shop Design	# of Best Solution
2-2	10	2-2-1	4	2-2-2	13	2-2-2-2-2	15	2-2-2-2-2-2-2-1	14
2-1-1	0	2-1-2	5	2-2-1-1	0	2-1-2-1-2-2	0	2-2-1-1-2-1-1-2-1-1-1	0
1-2-1	5	1-2-2	6	2-1-2-1	0	1-1-2-1-2-2-1	0	1-1-1-1-2-2-2-2-1-2	1
1-1-2	0	2-1-1-1	0	2-1-1-2	0	1-1-2-1-1-2-1-1	0	1-2-2-1-1-1-1-1-1-2-2	0
1-1-1-1	0	1-2-1-1	0	1-2-2-1	2	1-1-1-1-1-1-1-1-1-1	0	1-1-1-1-1-1-1-1-1-1-1-1-1-1-1	0
-		1-1-2-1	0	1-2-1-2	0	-		-	
-		1-1-1-2	0	1-1-2-2	0	-		-	
-		1-1-1-1-1	0	2-1-1-1-1	0	-		-	
-		-		1-2-1-1-1	0	-		-	
-		-		1-1-2-1-1	0	-		-	
-		-		1-1-1-2-1	0	-		-	
-		-		1-1-1-1-2	0	-		-	
-		-		1-1-1-1-1-1	0	-		-	
Total	15		15		15		15		15

CHAPTER 6

CASE STUDY AT ÖZEN COMPRESSOR COMPANY

In this chapter, we handle the flow shop scheduling problem of ÖZEN Compressor Company in Konya. Then we implement our proposed heuristic approaches for this problem and discuss the results.

6.1. General Information About ÖZEN Compressor

ÖZEN Compressor was founded in 1970 by its owner Mehmet Özen. The company's half a century of industrial machinery manufacturing experience starts with the production of welding machines and air compressors. Today, they manufacture and install compressed-air equipment at their state-of-the-art production plant, which is outfitted with world-class equipment. In addition to half a century of industrial experience that has now been passed on to the second generation, the company's innovative and customer-satisfaction-oriented work approach makes it possible to manufacture reliable, durable and efficient products.

Having the vision of "Producing efficient and durable compressed-air equipment with small carbon footprints that provide high air quality (3E rules: Efficient, Enduring, and Ecological)", the mission of the company is "To be a brand that produces reliable, accessible and competitive compressed-air equipment and that becomes solution partners for our customers in all businesses where compressed air is used. To have loyalty to company values, sustainable stability, and growth, both individually and as a team".

The company produces low- and medium-pressure piston and screw compressors, air dryers, air tanks, and air accessories. Being manufactured for all areas of life (automotive, chemical and petrochemical, textile, food, energy, health, logistics, glass industry, etc.), ÖZEN Compressor products address a wide range of industries.

ÖZEN Compressor, which prioritizes durability and productivity for its products, meets the compressed-air demands of small and large-sized customers from every industry in Turkey and the world. The company is one of the leading producers in the Eastern Europe and the Middle East. It has sales and service activities in 20 countries through 100 distributors, and exports to 45 countries constituting 30% of its sales.

In 2018, combining its half a century of industrial machinery manufacturing experience with foreign manpower and synergy, ÖZEN Compressor began assembling in Charlotte (NC) in the USA. Achieving rapid product delivery with one flow assembly line, the company is also standing by its customers in the USA with a wide range of spare parts and service support.

In recent years, especially with the entry to the USA market, air compressor demand has increased, and thus, the production capacity of ÖZEN Compressor has turned out to be inadequate. Hence, the firm has decided to improve the production system. In this study we intend to help increase the production rate of the plant and thus to increase the production capacity in order to meet the increasing demand.

6.2. Product Range and Production Capabilities

ÖZEN Compressor offers its customers a wide range of products with about 2000 alternatives. The main products are the piston compressors and screw compressors. Besides, air dryers, air tanks, and other air accessories are produced in the plant. The product range is shown in Figure 6.1, where each product has different pressure options, and so there are about 2000 total sku's.

Since it is the most crucial product group and the most congestion occurs at the manufacturing of this product group, we decide to deal with the screw compressor production.

On the other side, ÖZEN Compressor is able to execute many different production techniques in its own facility. We can categorize ÖZEN's production processes as follow:

- Sheet metal processing and painting
    - Laser cutting
    - Punch cutting
    - Sheet metal bending
    - Welding
    - Painting
- Machining
    - CNC Vertical Machining
    - CNC Horizontal Machining
    - CNC Turning
- Assembly
    - Pre-assembly
    - Piston compressor assembly line
    - Screw compressor assembly line
- Testing
    - Final check and test

When a screw compressor is put into production, first its metal components such as canopy and chassis are produced, and then they are painted. After that, pre-assembly processes are carried out, and the machine comes to the screw compressor assembly line. Finally, the test procedures of the machine are realized, and the machine is sent to the customer.

In the screw compressor production process, the assembly line appears as a bottleneck, and the volume of weekly screw compressor production is always determined by the assembly line performance. Therefore, we decide to focus on the screw compressor assembly line and implement our study here.

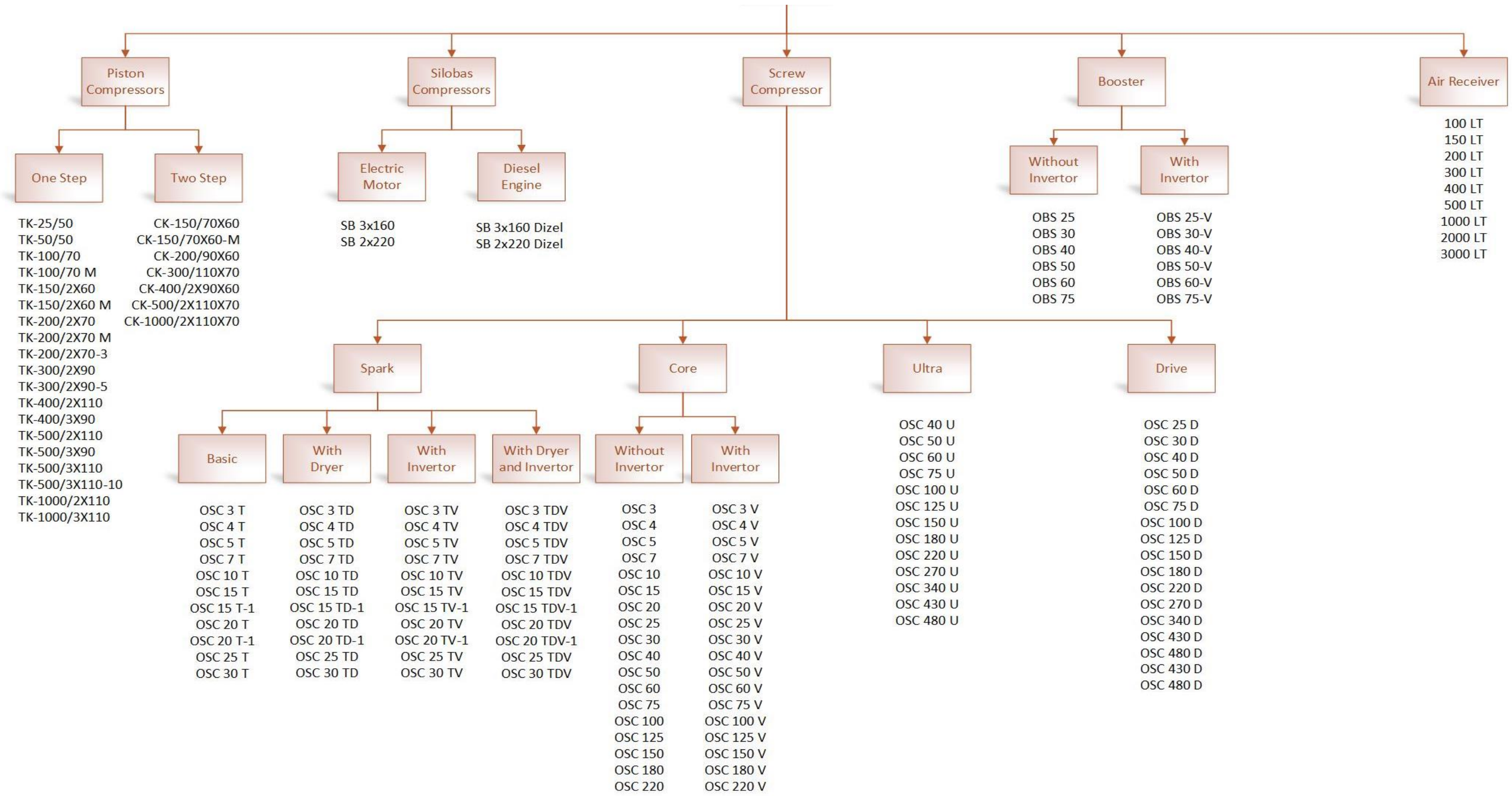

Figure 6.1 Product Range

6.3. Production Planning and Execution

At ÖZEN Compressor, production planning is carried out by the backward scheduling method. First, the machines that will enter the last assembly line are determined according to the due dates, and then the preliminary preparations of these machines are planned. The production schedule is planned on a weekly basis and one week in advance. For instance, the plan for the final assembly line for the next week is made earlier this week. When making this plan, the machines, which are produced, are determined according to their due dates. Then, necessary preliminary preparations and pre-assemblies are planned for this week. Thus, when the next week starts, all preparations of the machines (products) to be entered into the final assembly will have been completed.

In production planning, each model has a common due date that is the end of the week. Therefore, the aim here is to finish the production plan for that week as soon as possible. In other words, the aim is the minimization of makespan. In case a machine cannot be produced in the planned week, it is rescheduled to the beginning of next week.

In the weekly planning, it may be requested that any model can be produced in any quantity, or that one model may not be produced at all. Therefore, in general, there may be very different models in weekly planning, and after a weekly plan is composed, no changes are made, that is, it is frozen. In addition, since there is no setup time when moving from the production of one model to another, it is very easy to follow a mix-model production which is usually preferred by the production planning department. This situation allows us to implement our study to the company more easily.

Besides, we cannot talk about any flow in the screw compressor assembly line. An operator starts assembling a machine, does all the steps himself, and completes the assembly on a place where he starts at the beginning, that means, the production process is more like a project-type process in a fixed-position layout. There are five operators in the screw compressor assembly line. Although there is not much difference between these operators in terms of talent, there is a division of labor among themselves. In general, three of them are engaged in the assembly of machines of 60 hp and smaller, while the other two are engaged in the assembly of machines of 75 hp and bigger. However, depending on the models of the machines to be produced in that week, the number of operators in these two teams may vary. For example, if the majority of a weekly production plan consists of big machines, two operators pass to the other team and a total of four operators assemble the big machines where only one operator handles the small machines. In the implementation of our study, we consider these operators as multi-skilled operators and hence they can operate every task of products.

6.4. An Illustrative Example from ÖZEN Compressor Company

To observe the performance of our proposed heuristic approaches, we examine and experiment 4-week screw compressor manufacturing of ÖZEN Compressor. In this section, we study just 1stweek in detail and then share the results of the 2nd, 3rd, and 4th weeks. The details of the other weeks (weeks 2, 3, and 4) such as processing times and their schedules are given in appendices.

First of all, we handle the models, their operations, and processing times. There are about 1000 different types of machines in the screw compressor group. If we omit the pressure value options, we have to consider about 120 different types of screw compressors, which have almost the same final mounting operations but only the times vary because of the size differences of the assembled parts. For this reason, we group these 120 screw compressor models and reduce the number of types to 41. Grouped screw compressor models and their processing times are given in Table 6.1. We consider constant processing times for all models as indicated in Table 6.1, and each model possibly has a different number of operations to be completed. For instance, while the assembly of OSC 7 model is completed with 4 tasks, OSC 7 T model needs 5 tasks. Therefore, we use “-“sign in Table 6.1 where the task is not used for the model in question.

Furthermore, for each model, precedence relation is the same as indicated in Table 6.1. For example, task B for any model cannot start before task A is completed. Besides, since no task requires a special machine or special tool during operation, any task can be operated at any workstation if it assures the precedence relationship.

Table 6.1 Processing Times of Screw Compressor Models

Models-Processing Times (m)	**A***	**B**	**C**	**D**	**E**	**F**	**G**	**Total**
OSC 7 - OSC 7-V	40,0	26,0	14,0	-	44,0	-	-	124,0
OSC 7 T - OSC 7 VT	40,0	26,0	14,0	-	44,0	30,0	-	154,0
OSC 7 TD - OSC 7 VTD	40,0	26,0	14,0	-	44,0	30,0	50,0	204,0
OSC 10 - OSC 10-V	40,0	26,0	14,0	-	44,0	-	-	124,0
OSC 10 T - OSC 10 VT	40,0	26,0	14,0	-	44,0	30,0	-	154,0
OSC 10 TD - OSC 10 VTD	40,0	26,0	14,0	-	44,0	30,0	50,0	204,0
OSC 15 - OSC 15-V	46,0	28,0	14,0	-	48,0	-	-	136,0
OSC 15 T - OSC 15 VT - OSC 15 T-1 - OSC 15 VT-1	46,0	28,0	14,0	-	48,0	30,0	-	166,0
OSC 15 TD - OSC 15 VTD - OSC 15 TD-1 - OSC 15 VTD-1	46,0	28,0	14,0	-	48,0	30,0	54,0	220,0
OSC 20 - OSC 20-V	46,0	28,0	14,0	-	48,0	-	-	136,0
OSC 20 T - OSC 20 VT - OSC 20 T-1 - OSC 20 VT-1	46,0	28,0	14,0	-	48,0	30,0	-	166,0
OSC 20 TD - OSC 20 VTD - OSC 20 TD-1 - OSC 20 VTD-1	46,0	28,0	14,0	-	48,0	30,0	54,0	220,0
OSC 25 - OSC 25-V	52,0	34,0	16,0	-	60,0	-	-	162,0
OSC 25 T - OSC 25 VT	52,0	34,0	16,0	-	60,0	36,0	-	198,0
OSC 25 TD - OSC 25 VTD	52,0	34,0	16,0	-	60,0	36,0	60,0	258,0
OSC 30 - OSC 30-V	52,0	34,0	16,0	-	60,0	-	-	162,0
OSC 30 T - OSC 30 VT	52,0	34,0	16,0	-	60,0	36,0	-	198,0

Table 6.1 (continued)

OSC 30 TD - OSC 30 VTD	52,0	34,0	16,0	-	60,0	36,0	60,0	258,0
OSC 40 - OSC 40-V	58,0	34,0	16,0	-	60,0	-	-	168,0
OSC 40 T - OSC 40 VT	58,0	34,0	16,0	-	60,0	36,0	-	204,0
OSC 40 TD - OSC 40 VTD	58,0	34,0	16,0	-	60,0	36,0	60,0	264,0
OSC 50 - OSC 50-V	62,0	40,0	20,0	-	70,0	-	-	192,0
OSC 60 - OSC 60-V	62,0	40,0	20,0	-	70,0	-	-	192,0
OSC 75 - OSC 75-V	70,0	40,0	22,0	44,0	80,0	-	-	256,0
OSC 100 - OSC 100-V	70,0	44,0	28,0	48,0	90,0	-	-	280,0
OSC 125 - OSC 125-V	70,0	44,0	28,0	48,0	90,0	-	-	280,0
OSC 150 - OSC 150-V	80,0	96,0	32,0	50,0	110,0	-	-	368,0
OSC 180 - OSC 180-V	80,0	96,0	32,0	50,0	110,0	-	-	368,0
OSC 25 D	46,0	34,0	16,0	40,0	60,0	-	-	196,0
OSC 30 D	46,0	34,0	16,0	40,0	60,0	-	-	196,0
OSC 40 D - OSC 40 U	48,0	34,0	16,0	40,0	60,0	-	-	198,0
OSC 50 D - OSC 50 U	52,0	40,0	20,0	44,0	70,0	-	-	226,0
OSC 60 D - OSC 60 U	52,0	40,0	20,0	44,0	70,0	-	-	226,0
OSC 75 D - OSC 75 U	56,0	40,0	22,0	44,0	80,0	-	-	242,0
OSC 100 D - OSC 100 U	62,0	44,0	28,0	48,0	90,0	-	-	272,0
OSC 125 D - OSC 125 U	62,0	44,0	28,0	48,0	90,0	-	-	272,0
OSC 150 D - OSC 150 U	72,0	96,0	32,0	50,0	110,0	-	-	360,0
OSC 180 D - OSC 180 U	72,0	96,0	32,0	50,0	110,0	-	-	360,0
OSC 220 D - OSC 220 U	78,0	136,0	42,0	54,0	130,0	-	-	440,0
OSC 270 D - OSC 270 U	78,0	146.0	42,0	54,0	140,0	-	-	460,0
OSC 340 D - OSC 340 U	86,0	186,0	50,0	54,0	150,0	-	-	526,0

*A: Assembly of chassis, electric motor and air end on skid
B: Assembly of cooler system in canopy
C: Assembly of electric cabinet in canopy
D: Assembly of separator tank on skid
E: Assembly of filters/hoses
F: Assembly of compressor on air tank
G: Assembly of airdryer on air tank

Considering these conditions, we perform our study for 2 parallel flow shops with 5 operators. The job list that was produced in 1st week is given in Table 6.2 and the related processing times according to jobs and shop design are given in Table 6.3.

Table 6.2 Job List of the 1st Week

JOB NO	MODEL NAME	QUANTITY
J1	OSC 20 TD	1
J2	OSC 15 TD	1
J3	OSC 7 T	1
J4	OSC 20	1
J5	OSC 20 VTD-1	1
J6	OSC 30 VTD	1
J7	OSC 10 TD	1
J8	OSC 50	1
J9	OSC 150 D	1
J10	OSC 20	1
J11	OSC 30	1
J12	OSC 20 VTD-1	1
J13	OSC 60	1
J14	OSC 60 D	1
J15	OSC 15 TD	1
J16	OSC 40 U	1
J17	OSC 20 TD	1
J18	OSC 25 TD	1
J19	OSC 30 D	1
J20	OSC 30 D	1
J21	OSC 150 D	1
J22	OSC 50 V	1
J23	OSC 75 V	1

Table 6.3 Processing Times of Jobs for the 1st Week Production Scheduling

Shop Design	Processing Times (min)	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23
2-2-1	1st Stage	74	74	66	74	74	86	66	102	168	74	86	74	102	92	74	82	74	86	80	80	168	102	110
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	62	62	58	62	62	76	58	90	192	62	76	62	90	134	62	116	62	76	116	116	192	90	146
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	84	84	30	0	84	96	80	0	0	0	0	84	0	0	84	0	84	96	0	0	0	0	0
	Total	220	220	154	136	220	258	204	192	360	136	162	220	192	226	220	198	220	258	196	196	360	192	256
2-1-2	1st Stage	74	74	66	74	74	86	66	102	168	74	86	74	102	92	74	82	74	86	80	80	168	102	110
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	14	14	14	14	14	16	14	20	82	14	16	14	20	64	14	56	14	16	56	56	82	20	66
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	132	132	74	48	132	156	124	70	110	48	60	132	70	70	132	60	132	156	60	60	110	70	80
	Total	220	220	154	136	220	258	204	192	360	136	162	220	192	226	220	198	220	258	196	196	360	192	256
1-2-2	1st Stage	46	46	40	46	46	52	40	62	72	46	52	46	62	52	46	48	46	52	46	46	72	62	70
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	42	42	40	42	42	50	40	60	178	42	50	42	60	104	42	90	42	50	90	90	178	60	106
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	132	132	74	48	132	156	124	70	110	48	60	132	70	70	132	60	132	156	60	60	110	70	80
	Total	220	220	154	136	220	258	204	192	360	136	162	220	192	226	220	198	220	258	196	196	360	192	256

Table 6.3 (continued)

2-1-1-1	1st Stage	74	74	66	74	74	86	66	102	168	74	86	74	102	92	74	82	74	86	80	80	168	102	110
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	14	14	14	14	14	16	14	20	82	14	16	14	20	64	14	56	14	16	56	56	82	20	66
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	48	48	44	48	48	60	44	70	110	48	60	48	70	70	48	60	48	60	60	60	110	70	80
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	84	84	30	0	84	96	80	0	0	0	0	84	0	0	84	0	84	96	0	0	0	0	0
	Total	220	220	154	136	220	258	204	192	360	136	162	220	192	226	220	198	220	258	196	196	360	192	256
1-2-1-1	1st Stage	46	46	40	46	46	52	40	62	72	46	52	46	62	52	46	48	46	52	46	46	72	62	70
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	42	42	40	42	42	50	40	60	178	42	50	42	60	104	42	90	42	50	90	90	178	60	106
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	48	48	44	48	48	60	44	70	110	48	60	48	70	70	48	60	48	60	60	60	110	70	80
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	84	84	30	0	84	96	80	0	0	0	0	84	0	0	84	0	84	96	0	0	0	0	0
	Total	220	220	154	136	220	258	204	192	360	136	162	220	192	226	220	198	220	258	196	196	360	192	256
1-1-2-1	1st Stage	46	46	40	46	46	52	40	62	72	46	52	46	62	52	46	48	46	52	46	46	72	62	70
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	28	28	26	28	28	34	26	40	96	28	34	28	40	40	28	34	28	34	34	34	96	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	62	62	58	62	62	76	58	90	192	62	76	62	90	134	62	116	62	76	116	116	192	90	146
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.3 (continued)

	4th Stage	84	84	30	0	84	96	80	0	0	0	0	84	0	0	84	0	84	96	0	0	0	0	0
	Total	220	220	154	136	220	258	204	192	360	136	162	220	192	226	220	198	220	258	196	196	360	192	256
1-1-1-2	1st Stage	46	46	40	46	46	52	40	62	72	46	52	46	62	52	46	48	46	52	46	46	72	62	70
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	28	28	26	28	28	34	26	40	96	28	34	28	40	40	28	34	28	34	34	34	96	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	14	14	14	14	14	16	14	20	82	14	16	14	20	64	14	56	14	16	56	56	82	20	66
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	132	132	74	48	132	156	124	70	110	48	60	132	70	70	132	60	132	156	60	60	110	70	80
	Total	220	220	154	136	220	258	204	192	360	136	162	220	192	226	220	198	220	258	196	196	360	192	256
1-1-1-1-1	1st Stage	46	46	40	46	46	52	40	62	72	46	52	46	62	52	46	48	46	52	46	46	72	62	70
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	28	28	26	28	28	34	26	40	96	28	34	28	40	40	28	34	28	34	34	34	96	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	14	14	14	14	14	16	14	20	82	14	16	14	20	64	14	56	14	16	56	56	82	20	66
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	48	48	44	48	48	60	44	70	110	48	60	48	70	70	48	60	48	60	60	60	110	70	80
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5th Stage	84	84	30	0	84	96	80	0	0	0	0	84	0	0	84	0	84	96	0	0	0	0	0
	Total	220	220	154	136	220	258	204	192	360	136	162	220	192	226	220	198	220	258	196	196	360	192	256

In Section 6.4.1 the heuristic approach solutions of the 1st-week experiment are given, whereas the real execution by ÖZEN Compressor is shown in Section 6.4.2. Finally, the analysis of the whole 4-week production is summarized in Section 6.4.3.

6.4.1. Heuristic Approach Solutions

We use our four heuristic approaches, SPT&NEH, LPT&NEH, SPT, and LPT, to find out the best solution. As a result of each heuristic method, job scheduling and the related C_{max} values are given in the following tables (Tables 6.4 - 6.12).

Table 6.4 Heuristic Results with 2-2-1 Shop Design

SHOP DESIGN		**2-2-1**		
Heuristic Method	Solution Alternative	Job Schedule		C_{max}
SPT&NEH Algorithm	Alternative 1	FS1	J12-J17-J20-J02-J18-J23-J21-J08-J22-J07-J03-J04	1216
		FS2	J19-J15-J13-J14-J01-J06-J11-J05-J10-J16-J09	
LPT&NEH Algorithm	Alternative 1	FS1	J07-J06-J19-J12-J23-J02-J17-J09-J08-J22-J04	1216
		FS2	J15-J03-J16-J18-J14-J01-J05-J20-J21-J13-J11-J10	
	Alternative 2	FS1	J07-J06-J19-J12-J23-J02-J17-J09-J08-J22-J04	
		FS2	J15-J03-J16-J18-J14-J01-J05-J20-J21-J13-J10-J11	
	Alternative 3	FS1	J07-J06-J19-J12-J23-J02-J17-J09-J08-J22-J04	
		FS2	J15-J03-J16-J18-J14-J01-J05-J20-J21-J10-J13-J11	
SPT	Alternative 1	FS1	J04-J03-J08-J22-J16-J01-J05-J15-J14-J06-J09	1384
		FS2	J10-J11-J13-J19-J20-J07-J02-J12-J17-J23-J18-J21	
LPT	Alternative 1	FS1	J21-J18-J14-J01-J05-J15-J07-J19-J08-J22-J03-J10	1468
		FS2	J09-J06-J23-J02-J12-J17-J16-J20-J13-J11-J04	

As seen in Table 6.4, LPT&NEH Algorithm comes up with three solution alternatives that give 1216 minutes as an objective value. The difference between these three solution alternatives is the position of J10 in the job sequence of flow shop 2. In Alternative 1, J10 is positioned at the 12th place, whereas it enters the FS2 at the 11th and 10th place respectively in Alternative 2 and Alternative 3.

We give the solution alternatives in Table 6.4 for 2-2-1 design. However, we indicate only one solution alternative for each heuristic for the other design alternatives.

Table 6.5 Heuristic Results with 2-1-2 Shop Design

SHOP DESIGN				2-1-2
Heuristic Method	Solution Alternative	Job Schedule		C_{max}
SPT&NEH Algorithm	Alternative 1	FS1	J01-J16-J03-J05-J08-J22-J04-J15-J06-J09-J14	1266
		FS2	J07-J02-J13-J19-J12-J20-J11-J10-J17-J23-J18-J21	
LPT&NEH Algorithm	Alternative 1	FS1	J05-J06-J01-J19-J09-J15-J23-J08-J22-J03-J16-J10	1218
		FS2	J07-J02-J12-J21-J18-J14-J17-J20-J04-J13-J11	
SPT	Alternative 1	FS1	J04-J03-J08-J22-J16-J01-J05-J15-J14-J06-J09	1416
		FS2	J10-J11-J13-J19-J20-J07-J02-J12-J17-J23-J18-J21	
LPT	Alternative 1	FS1	J21-J18-J14-J01-J05-J15-J07-J19-J08-J22-J03-J10	1454
		FS2	J09-J06-J23-J02-J12-J17-J16-J20-J13-J11-J04	

In 2-1-2 shop design, LPT&NEH Algorithm offers us two more solution alternatives where J10 enters the FS1 in the 10^{th} and 11^{th} places.

Table 6.6 Heuristic Results with 1-2-2 Shop Design

SHOP DESIGN				1-2-2
Heuristic Method	Solution Alternative	Job Schedule		C_{max}
SPT&NEH Algorithm	Alternative 1	FS1	J17-J03-J12-J08-J02-J20-J07-J18-J21-J23-J22-J04	1344
		FS2	J11-J05-J15-J19-J01-J16-J06-J09-J14-J13-J10	
LPT&NEH Algorithm	Alternative 1	FS1	J12-J06-J02-J18-J09-J17-J16-J19-J08-J22-J04-J10	1286
		FS2	J01-J05-J15-J21-J14-J07-J23-J20-J13-J03-J11	
SPT	Alternative 1	FS1	J04-J03-J08-J22-J19-J16-J01-J05-J14-J23-J06-J18	1484
		FS2	J10-J11-J13-J20-J07-J02-J12-J15-J19-J09-J21	
LPT	Alternative 1	FS1	J21-J23-J01-J02-J07-J19-J08-J03	1438
		FS2	J09-J06-J18-J14-J05-J12-J15-J17-J16-J20-J13-J22-J11-J04-J10	

In 1-2-2 shop design, LPT&NEH Algorithm gives us six more solution alternatives. The only change in these alternatives is J10's place in the sequence that is 6, 7, 8, 9, 10 and 11th place in FS1.

Table 6.7 Heuristic Results with 2-1-1-1 Shop Design

SHOP DESIGN			**2-1-1-1**	
Heuristic Method	Solution Alternative	Job Schedule		C_{max}
SPT&NEH Algorithm	Alternative 1	FS1	J03-J04-J15-J08-J22-J07-J01-J05-J19-J14-J18-J21-J16-J06-J23	1508
		FS2	J10-J11-J17-J13-J02-J12-J20-J09	
LPT&NEH Algorithm	Alternative 1	FS1	J06-J11-J07-J14-J18-J05-J12-J15-J09-J16-J19-J20-J08-J22-J04-J10	1500
		FS2	J17-J23-J02-J21-J01-J13-J03	
SPT	Alternative 1	FS1	J04-J03-J08-J22-J20-J07-J02-J12-J17-J23-J18-J21	1586
		FS2	J10-J11-J13-J19-J16-J01-J05-J15-J14-J06-J09	
LPT	Alternative 1	FS1	J21-J18-J23-J17-J12-J02-J07-J20-J22-J08-J03-J04	1762
		FS2	J09-J06-J14-J15-J05-J01-J16-J19-J13-J11-J10	

In 2-1-1-1 shop design, SPT&NEH Algorithm finds out five more solution alternatives in which the position of J21 in the sequence in FS1 varies as 7, 8, 9, 10 and 11. Besides, there are ten more solution alternatives provided by LPT&NEH Algorithm, and the position of J10 in the sequence in FS1 changes as 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15.

In 1-2-1-1 shop design, SPT&NEH Algorithm reaches one more solution alternative, and the position of J21 in the sequence in FS2 is 7 in that solution alternative. Furthermore, LPT&NEH Algorithm offers nine more solution alternatives. In six of them J10 enters to FS1 at the 7, 8, 9, 10, 11 and 12th place, and in the remaining three alternatives J10 enters to FS2 at the 9, 10 and 11th place.

Table 6.8 Heuristic Results with 1-2-1-1 Shop Design

SHOP DESIGN			**1-2-1-1**	
Heuristic Method	Solution Alternative	Job Schedule		C_{max}
SPT&NEH Algorithm	Alternative 1	FS1	J03-J04-J11-J05-J20-J08-J17-J18-J13-J12-J14-J15-J06-J22-J19	1530
		FS2	J10-J07-J01-J02-J16-J23-J09-J21	
LPT&NEH Algorithm	Alternative 1	FS1	J15-J11-J07-J20-J13-J09-J23-J05-J18-J02-J14-J04-J10	1532
		FS2	J17-J16-J08-J22-J03-J06-J01-J12-J19-J21	
SPT	Alternative 1	FS1	J04-J03-J08-J22-J20-J07-J01-J05-J15-J14-J06-J18-J21	1652
		FS2	J10-J11-J13-J19-J16-J02-J12-J17-J23-J09	
LPT	Alternative 1	FS1	J21-J23-J01-J02-J12-J17-J16-J20-J13-J11-J04	1782
		FS2	J09-J06-J18-J14-J05-J15-J07-J19-J08-J22-J03-J10	

Table 6.9 Heuristic Results with 1-1-2-1 Shop Design

SHOP DESIGN			**1-1-2-1**	
Heuristic Method	Solution Alternative	Job Schedule		C_{max}
SPT&NEH Algorithm	Alternative 1	FS1	J09-J08-J17-J22-J12-J20-J18-J23-J03-J07-J02-J04	1352
		FS2	J19-J21-J01-J14-J15-J06-J16-J05-J13-J11-J10	
LPT&NEH Algorithm	Alternative 1	FS1	J09-J18-J01-J02-J12-J15-J08-J22-J17-J07-J03-J10-J23	1382
		FS2	J05-J06-J20-J14-J21-J13-J11-J04-J16-J19	
SPT	Alternative 1	FS1	J04-J03-J08-J22-J19-J16-J01-J05-J14-J23-J06-J18	1508
		FS2	J10-J11-J13-J20-J07-J02-J12-J15-J19-J09-J21	
LPT	Alternative 1	FS1	J21- J23-J01-J02- J07 -J19-J08-J03	1488
		FS2	J09-J06-J18-J14-J05-J12-J15-J17-J16-J20-J13-J22-J11-J04-J10	

In 1-1-2-1 shop design, LPT&NEH Algorithm provides seventeen more solution alternatives. In nine of them J10 enters to FS1 at the 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12th place, and in the remaining eight alternatives J10 enters to FS2 at the 1, 2, 6, 7, 8, 9, 10 and 11th place.

Table 6.10 Heuristic Results with 1-1-1-2 Shop Design

SHOP DESIGN			**1-1-1-2**	
Heuristic Method	Solution Alternative	Job Schedule		C_{max}
SPT&NEH Algorithm	Alternative 1	FS1	J17-J03-J02-J08-J07-J12-J20-J18-J21-J23-J22-J04	1364
		FS2	J11-J15-J06-J19-J01-J05-J09-J16-J14-J13-J10	
LPT&NEH Algorithm	Alternative 1	FS1	J07-J06-J05-J15-J09-J01-J23-J08-J19-J22-J03-J10	1286
		FS2	J18-J02-J12-J17-J20-J21-J14-J13-J16-J11-J04	
SPT	Alternative 1	FS1	J04-J03-J08-J22-J20-J07-J01-J05-J15-J14-J06-J18-J21	1508
		FS2	J10-J11-J13-J19-J16-J02-J12-J17-J23-J09	
LPT	Alternative 1	FS1	J21-J23-J01-J02-J12-J17-J16-J20-J13-J11-J04	1440
		FS2	J09-J06-J18-J14-J05-J15-J07-J19-J08-J22-J03-J10	

In 1-1-2-1 shop design, there are no alternative solutions.

In 1-1-1-1-1 shop design, SPT&NEH Algorithm finds out one more solution alternative where J21 enters to FS1 at the 20th place in the sequence. In addition, LPT&NEH Algorithm comes up with fifteen more solution alternatives. In eight of them J10 enters to FS1 at the 8, 9, 10, 11, 12, 13, 14 and 15th place, and in the remaining seven alternatives J10 enters to FS2 at the 2, 3, 4, 5, 6, 7 and 8th place.

We apply the same steps for the other weeks as we followed for the 1st week and share their details in the appendices section. The completion times of all weeks found by the heuristic approaches are summarized in Table 6.12.

Table 6.11 Heuristic Results with 1-1-1-1-1 Shop Design

SHOP DESIGN		**1-1-1-1-1**			
Heuristic Method	Solution Alternative	Job Schedule			C_{max}
SPT&NEH Algorithm	Alternative 1	FS1	J12-J03-J05-J04-J02-J11-J07-J01-J08-J13-J22-J19-J17-J20-J16-J06-J18-J14-J23-J09-J21		1522
		FS2	J15-J10		
LPT&NEH Algorithm	Alternative 1	FS1	J06-J23-J01-J05-J19-J15-J20-J07-J09-J08-J13-J22-J03-J11-J04-J10		1572
		FS2	J18-J17-J14-J02-J12-J16-J21		
SPT	Alternative 1	FS1	J04-J03-J08-J22-J20-J07-J02-J12-J17-J23-J18-J21		1604
		FS2	J10-J11-J13-J19-J16-J01-J05-J15-J14-J06-J09		
LPT	Alternative 1	FS1	J21-J18-J23-J17-J12-J02-J07-J20-J22-J08-J03-J04		1782
		FS2	J09-J06-J14-J15-J05-J01-J16-J19-J13-J11-J10		

Table 6.12 Completion Times of Four Weeks in the Planning Horizon

Weeks	**Shop Design**	**SPT&NEH Algorithm**	**LPT&NEH Algorithm**	**SPT**	**LPT**
Week 1	2-2-1	1216	1216	1384	1468
	2-1-2	1266	1218	1416	1454
	1-2-2	1344	1286	1484	1438
	2-1-1-1	1508	1500	1586	1762
	1-2-1-1	1530	1532	1652	1782
	1-1-2-1	1352	1382	1508	1488
	1-1-1-2	1364	1286	1508	1440
	1-1-1-1-1	1522	1572	1604	1782
Week 2	2-2-1	1304	1188	1484	1296
	2-1-2	1230	1180	1390	1266
	1-2-2	1392	1392	1596	1392
	2-1-1-1	1572	1550	1736	1688
	1-2-1-1	1614	1590	1700	1736

Table 6.12 (continued)

	1-1-2-1	1404	1392	1650	1392
	1-1-1-2	1410	1392	1596	1392
	1-1-1-1-1	1596	1596	1736	1820
Week 3	2-2-1	1406	1342	1468	1526
	2-1-2	1366	1266	1468	1504
	1-2-2	1502	1402	1694	1454
	2-1-1-1	1646	1646	1646	1886
	1-2-1-1	1772	1834	1700	1942
	1-1-2-1	1552	1504	1754	1542
	1-1-1-2	1540	1402	1754	1532
	1-1-1-1-1	1754	1726	1754	1942
Week 4	2-2-1	1526	1434	1574	1690
	2-1-2	1428	1400	1526	1634
	1-2-2	1422	1422	1658	1560
	2-1-1-1	1682	1770	1716	1900
	1-2-1-1	1690	1770	1736	1966
	1-1-2-1	1640	1556	1736	1722
	1-1-1-2	1510	1478	1736	1648
	1-1-1-1-1	1730	1704	1736	1966

6.4.2. Real Execution in ÖZEN Compressor

As we mentioned in the above section, the screw compressor assembly line operators are divided into two teams at the ÖZEN Compressor Company. In general, Team 1 is composed of 3 operators and deal with the assembly of compressors of 60 hp and smaller, while Team 2 consists of 2 operators and focuses on the assembly of compressors of 75 hp and bigger. However, the number of operators in the teams may have changed with the total number of operators 5 to balance the workload in the four weeks that we have analyzed.

In light of this information, the production scheduling and its execution are as follows (Table 6.13).

Table 6.13 Production Schedules and Completion Times in ÖZEN Compressor

Weeks	Team 1	Team 2	C_{max} (min)
Week 1	J01-J02-J03-J04-J05-J06-J07-J08-J10-J11-J12-J13-J14-J15-J16-J17-J18-J19-J20-J22	J09-J21-J23	2565
Week 2	J01-J02-J03-J04-J05-J06-J07-J09-J10-J11-J12-J13-J14-J15-J16-J17-J18-J19-J20-J21-J22-J23	J08	2160
Week 3	J01-J02-J03-J04-J06-J08-J09-J10-J11-J14-J15-J16-J17-J18-J19-J20-J22	J05-J07-J12-J13-J21-J23	2835
Week 4	J02-J03-J04-J07-J08-J13-J14-J15-J16-J17	J01-J05-J06-J09-J10-J11-J12-J18-J19	3240

6.4.3. Evaluation of Results

The comparison of the completion time of weekly production plans is indicated below in Table 6.14. In the table, only the best solution found by heuristic methods for each week is indicated. Furthermore, the heuristic method which finds the best solution and the shop design that the best solution is reached with are reported. Finally, the reduction in C_{max} values observed at ÖZEN Compressor for four weeks compared to the C_{max} values obtained with the heuristic methods is given in Table 6.14.

Table 6.14 Comparison of Completion Times of the Weekly Production Plan

Weeks	Number of Jobs	Total Work Content (min)	C_{max} (min) ÖZEN Schedule	C_{max} (min) Heuristic Schedule	Heuristic Method	Shop Design	Change in C_{max} (%)
Week 1	23	4996	2565	1216	SPT&NEH LPT&NEH	2-2-1	-52.6
Week 2	26	4750	2160	1180	LPT&NEH	2-1-2	-45.4
Week 3	23	5166	2835	1266	LPT&NEH	2-1-2	-55.3
Week 4	19	5432	3240	1400	LPT&NEH	2-1-2	-56.8

As can be seen from the table, heuristic solutions are superior to the ÖZEN Compressor schedule in each week. The average gap of four weeks is -52.5%, and that means if our proposed parallel flow shops and the scheduling method proposed for that shop design are implemented at ÖZEN Compressor, the production rate of the screw compressors is almost doubled at least.

CHAPTER 7

CONCLUSION

In this thesis, the parallel flow shop scheduling problem with common workstations is studied. In this flow shop, common workstations are preferred at some of the stages; limited buffer areas are placed between stages, and job crossing between flow shops is not allowed. The objective is to minimize the makespan. We also aim to find out the best shop design that provides the best makespan for a required workforce level, defined by the number of operators.

The production system at ÖZEN Compressor Company in Konya encourages us to conduct this study.

Since our problem is similar to both flexible flow shop scheduling and parallel flow shop scheduling, we develop two mathematical models separately, as Mixed Integer Linear Programming (MILP) models: FFS based MILP model that we are inspired by the mathematical model of Sawik (2000), and PFS based MILP model that we are inspired by Ribas (2017).

Besides, we present heuristic algorithms as alternative solution methods by which we aim to obtain optimal or near-optimal results at least in a short computational time. These approaches are modified from the well-known NEH Algorithm, and we call them as LPT&NEH and SPT&NEH which are the implementations of the NEH Algorithm to two parallel flow shops. SPT&NEH algorithm uses the non-decreasing ordering of total job processing times, while LPT&NEH algorithm uses the non-increasing ordering of total job processing times. Furthermore, we propose the two commonly used dispatching rules, SPT and LPT, to observe the efficiency and effectiveness of MILP models and the proposed NEH algorithms.

In the experiments that we execute with different sizes depending on the number of operators and jobs, we limit MILP’s run time with one hour because of the fact that our problems are large-sized, and we want to reach the results in a plausible time. The experimental results show that MILPs usually give the best solution, not the optimal due to the running time limit, and PFS based MILP model achieves better solutions than FFS based MILP model.

On the other hand, our proposed heuristics give outstanding results in the experiments. While they are superior to the dispatching rules, they reach very close results to the PFS based MILP with small gap values, especially in the experiments with a higher number of jobs. Moreover, as we intended, heuristic methods provide solutions in a very short time. While we can achieve the results in about one hour with the MILP model, we can obtain the solutions in under one-minute time with the proposed heuristic approaches. Even when there are 15 jobs to be processed, it takes less than one second to obtain solutions with the proposed heuristics.

According to the experiments in ÖZEN Compressor Company, our heuristic methods offer a doubling of production rate and make it possible to produce the same job orders in less than half the time it normally takes in the company. This improvement is exactly what the company wants and even more.

To the best of our knowledge, our thesis is the first study in the literature to solve parallel flow shop scheduling with common workstations problem with the aim of minimizing the makespan that considers the shop designs depending solely on the number of operators. We hope that our study may inspire further researches in the flow shop scheduling literature. Some parts of our study are open to improvement, and thus, sophisticated studies can be conducted by focusing on these parts, which are listed below:

- Processing times can be considered as stochastic such as operator-dependent processing times or learning-curve effects taken into account.
- Sequence-dependent setup times can be considered.
- Workstation restrictions can be considered since some operations of some jobs may require some certain special tools or machines.

- Job priorities, according to due dates, can be included.

- The number of parallel flow shops can be increased.

- Non-permutation flow shop scheduling can be considered.

REFERENCES

Agnetis, A., & Arbib, C. (1997). Concurrent operations assignment and sequencing for particular assembly problems in flow lines. *Annals of Operations Research 69: 1–31*.

Al-Salem, A. (2004). A Heuristic to minimize makespan in proportional parallel flow shops. *International Journal of Computing & Information Sciences, 2 (2), 98–107.*

Almeder, C., & Hartl, R.F. (2013). A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. *International Journal of Production Economics145:88–95.*

Azizoğlu, M., Çakmak, E., & Kondakcı, S. (2001). A flexible flow shop problem with total flow time minimization. *European Journal of Operational Research 132 : 528-538.*

Bai, D., Liang, J., Liu, B., Tang, M., & Zhang, Z. (2017). Permutation flow shop scheduling problem to minimize nonlinear objective function with release dates. *Computers & Industrial Engineering 112: 336–347.*

Brah, S.A., & Wheeler, G.E. (1998). Comparison of scheduling rules in a flow shop with multiple processors: a simulation. *Simulation: Transactions of The Society for Modeling and Simulation International 71:302-311.*

Brah, S.A., & Loo, L.L. (1999). Heuristics for scheduling in a flow shop with multiple processors. *European Journal of Operational Research, 113(1):113–122.*

Campbell, H.G., Dudek, R. A., & Smith.M.L. (1970). A heuristic algorithm for the n job, m machine sequencing problem. *Management Science, 16(10), 630–637.*

Chen, B. (1995). Analysis of classes of heuristics for scheduling a two–stage flow–shop with parallel machines at one stage. *Journal of the Operational Research Society, 46(2):234–244.*

Dessouky, M.M., Dessouky, M.I., & Verma, S.K. (1998). Flowshop scheduling with identical jobs and uniform parallel machines. *European Journal of Operational Research 109 : 620-631.*

Dios, M., Fernandez-Viagas, V., & Framinan, J.M. (2018). Efficient heuristics for the hybrid flow shop scheduling problem with missing operations. *Computers & Industrial Engineering 115 : 88–99.*

Dong, J., Tong, W., Luo, T., Wang, X., Hu, J., Xu, Y., & Lin, G. (2017). An FPTAS for the parallel two-stage flow shop problem. *Theoretical Computer Science 657 : 64–72.*

Engin, O., Ceran, G., & Yilmaz M.K. (2011). An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems. *Applied Soft Computing, 11(3):3056–3065.*

Garey, M.R., Johnson, D.S., & Sethi, R.(1976). The complexity of flow shop and job shop scheduling. *Mathematics of Operations Research 1 (2), 117–129.*

Grabowski, J., & Pempera, J. (2000). Sequencing of jobs in some production system. *European Journal of Operational Research, 125(3):535–550.*

Grangeon, N., Tanguy, A., & Tchernev, N. (1999). Generic simulation model for hybrid flow-shop. *Computers &Industrial Engineering 37 : 207-210.*

Gupta, J.N.D. (1988). Two–stage, hybrid flow shop scheduling problem. *Journal of the Operational Research Society, 39(4):359–364.*

Gupta, J.N.D., Hariri, A.M.A., & Potts, C.N. (1997). Scheduling a two–stage hybrid flow shop with parallel machines at the first stage. *Annals of Operations Research, 69:171–191.*

Gupta, J.N.D., Kruger, K., Lauff, V., Werner, F., & Sotskov, Y.N. (2002). Heuristics for hybrid flow shops with controllable processing times and assignable due dates. *Computers & Operations Research, 29(10):1417–1439, 2002.*

Haouari, M., & M'Hallah, R. (1997). Heuristic algorithms for the two-stage hybrid flowshop problem. *Operations Research Letters 21 : 43-53.*

Hojati, M. (2016). Minimizing makespan in 2-stage disassembly flow shop scheduling problem. *Computers & Industrial Engineering 94 : 1–5.*

Hossain, S.,Asadujjaman, Nayon, A.A., & Bhattacharya, P. (2014). Minimization of makespan in flow shop scheduling using heuristics. *International Conference on Mechanical, Industrial and Energy Engineering.*

Kazemi, H., Mazdeh, M.M., & Rostami, M. (2017). The two stage assembly flow-shop scheduling problem with batching and delivery. *Engineering Applications of Artificial Intelligence63: 98–107.*

Khorasanian, D., & Moslehi, G. (2017). Two-machine flow shop scheduling problem with blocking, multi-task flexibility of the first machine, and preemption. *Computers & Operations Research 79 : 94–108.*

Kis, T., & Pesch, E.(2005).A review of exact solution methods for the non-preemptive multiprocessor flowshop problem. *European Journal of Operational Research 164 (3), 592–608.*

Kochhar, S., & Morris, R. J. T. (1987). Heuristic methods for flexible flow line scheduling. *Journal of Manufacturing Systems, 6(4):299–314.*

Kyparisis, G.J., & Koulamas, C. (2006). Flexible flow shop scheduling with uniform parallel machines. *European Journal of Operational Research 168 : 985–997.*

Laribi, I., Yalaoui, F., Belkaid, F., & Sari, Z. (2016). Heuristics for solving flow shop scheduling problem under resources constraints. *IFAC-Papers On Line 49-12 : 1478–1483.*

Li, J., & Pan, Q. (2015). Solving the large–scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. *Information Sciences, 316:487–502.*

Liao, C.J., Tjandradjaja, E., & Chung, T.P. (2012). An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem. *Applied Soft Computing, 12(6):1755–1764.*

Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: a survey. *Computers & Industrial Engineering 37 : 57-61.*

Marichelvam, M.K., Prabaharan, T., & Yang, X.S. (2014). Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. *Applied Soft Computing, 19:93–101.*

Nawaz, M., Enscore, E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *The International Journal of Management Science 11, 91-95.*

Nowicki, E., & Smutnicki, C. (1998). The flow shop with parallel machines: A tabu search approach. *European Journal of Operational Research, 106(2–3):226–253.*

Quadt, D., & Khun, H. (2007). A taxonomy of flexible flow line scheduling procedures. *European Journal of Operational Research 178 : 686–698.*

Rajendran, C., & Chaudhuri, D. (1992). A multistage parallel–processor flowshop problem with minimum flowtime. *European Journal of Operational Research, 57(1):111–122.*

Ribas, I., Companys, R., & Tort-Martorell, X. (2017). Efficient heuristics for the parallel blocking flow shop scheduling problem. *Expert Systems With Applications 74 : 41–54.*

Rossit, D.A., Tohme, F., & Frutos, M. (2018). The Non-Permutation Flow-Shop scheduling problem: A literature review. *The International Journal of Management Science, 77, 143-153.*

Ruiz, R., Serifoglu, F.S., & Urlings T. (2008).Modeling realistic hybrid flexible flowshop scheduling problems. *Computers & Operations Research, 35(4):1151–1175.*

Ruiz, R., & Vázquez-Rodríguez, J.A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research, 205, 1-18.*

Sawik, T. (2000). Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modeling, 31, 39-52.*

Schuurman, P., & Woeginger, G.J. (2000). A polynomial time approximation schemefor the two-stage multiprocessor flow shop problem. *Theoretical Computer Science 237 : 105-122.*

Semanco, P., & Modrak, V. (2012). A comparison of constructive heuristics with the objective of minimizing makespan in the flow shop scheduling problem. *Acta Polytechnica Hungarica 9(5):177-190.*

Soewandi, H., & Elmaghraby, S.E. (2001). Sequencing three-stage flexible flowshops with identical machines to minimize makespan. *IIE Transactions 33, 985-993.*

Sundararaghavan, P.S., Kunnathur, A.S., & Viswanathan, I. (1997). Minimizing makespan in parallel flowshops. *Journal of the Operational Research Society 48, 834-842.*

Taillard,E.(1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research, 47, 65–74.*

Tavakkoli-Moghaddam, R., Safaei, N., & Sassani, F. (2009). A memetic algorithm for the flexible flow line scheduling problem with processor blocking. *Computers & Operations Research 36:402 – 414.*

Tong, W., Miyano, E., Goebel, R., & Lin, G. (2018). An approximation scheme for minimizing the makespan of the parallel identical multi-stage flowshops. *Theoretical Computer Science 734, 24-31.*

Wardono, B., & Fathi, Y. (2004). A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities. *European Journal of Operational Research 155 : 380–401.*

Wittrock, R.J. (1985). Scheduling algorithms for flexible flow lines. *IBM Journal of Research and Development, 29(4):401–412.*

Wu, Y., Liu, M., & Wu, C. (2003). A genetic algorithm for solving flow shop scheduling problems with parallel machine and special procedure constraints. *Proceedings of the Second International Conference on Machine Learning and Cybernetics, vol. 3, IEEE Press, 1774-1779.*

Zhang, X., & Velde, S.V.D. (2012). Approximation algorithms for the parallel flow shop problem. *European Journal of Operational Research 216 : 544–552.*

APPENDIX

Table A.1 Job List of the 2nd Week of Case Study

JOB NO	MODEL NAME	QUANTITY
J1	OSC 30	1
J2	OSC 20 TD	1
J3	OSC 50	1
J4	OSC 50	1
J5	OSC 50	1
J6	OSC 50	1
J7	OSC 50	1
J8	OSC 180 U	1
J9	OSC 50	1
J10	OSC 40 U	1
J11	OSC 7 TD	1
J12	OSC 10	1
J13	OSC 10TD	1
J14	OSC 30	1
J15	OSC 15	1
J16	OSC 10 TD	1
J17	OSC 20	1
J18	OSC 40	1
J19	OSC 40	1
J20	OSC 15	1
J21	OSC 10	1
J22	OSC 20 TD-1	1
J23	OSC 15 TD-1	1
J24	OSC 10 TD	1
J25	OSC 7	1
J26	OSC 7	1

Table A.2 Processing Times of Jobs for the 2nd Week Production Scheduling of Case Study

Shop Design	Processing Times (min)	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	J25	J26
2-2-1	1st Stage	86	74	102	102	102	102	102	168	102	82	66	66	66	86	74	66	74	92	92	74	66	74	74	66	66	66
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	76	62	90	90	90	90	90	192	90	116	58	58	58	76	62	58	62	76	76	62	58	62	62	58	58	58
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	0	84	0	0	0	0	0	0	0	0	80	0	80	0	0	80	0	0	0	0	0	84	84	80	0	0
	Total	162	220	192	192	192	192	192	360	192	198	204	124	204	162	136	204	136	168	168	136	124	220	220	204	124	124
2-1-2	1st Stage	86	74	102	102	102	102	102	168	102	82	66	66	66	86	74	66	74	92	92	74	66	74	74	66	66	66
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	16	14	20	20	20	20	20	82	20	56	14	14	14	16	14	14	14	16	16	14	14	14	14	14	14	14
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	60	132	70	70	70	70	70	110	70	60	124	44	124	60	48	124	48	60	60	48	44	132	132	124	44	44
	Total	162	220	192	192	192	192	192	360	192	198	204	124	204	162	136	204	136	168	168	136	124	220	220	204	124	124
1-2-2	1st Stage	52	46	62	62	62	62	62	72	62	48	40	40	40	52	46	40	46	58	58	46	40	46	46	40	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	50	42	60	60	60	60	60	178	60	90	40	40	40	50	42	40	42	50	50	42	40	42	42	40	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	60	132	70	70	70	70	70	110	70	60	124	44	124	60	48	124	48	60	60	48	44	132	132	124	44	44
	Total	162	220	192	192	192	192	192	360	192	198	204	124	204	162	136	204	136	168	168	136	124	220	220	204	124	124
2-1-1-1	1st Stage	86	74	102	102	102	102	102	168	102	82	66	66	66	86	74	66	74	92	92	74	66	74	74	66	66	66
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	16	14	20	20	20	20	20	82	20	56	14	14	14	16	14	14	14	16	16	14	14	14	14	14	14	14
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A.2 (continued)

	3rd Stage	60	48	70	70	70	70	70	110	70	60	44	44	44	60	48	44	48	60	60	48	44	48	48	44	44	44
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	0	84	0	0	0	0	0	0	0	0	80	0	80	0	0	80	0	0	0	0	0	84	84	80	0	0
	Total	162	220	192	192	192	192	192	360	192	198	204	124	204	162	136	204	136	168	168	136	124	220	220	204	124	124
1-2-1-1	1st Stage	52	46	62	62	62	62	62	72	62	48	40	40	40	52	46	40	46	58	58	46	40	46	46	40	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	50	42	60	60	60	60	60	178	60	90	40	40	40	50	42	40	42	50	50	42	40	42	42	40	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	60	48	70	70	70	70	70	110	70	60	44	44	44	60	48	44	48	60	60	48	44	48	48	44	44	44
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	0	84	0	0	0	0	0	0	0	0	80	0	80	0	0	80	0	0	0	0	0	84	84	80	0	0
	Total	162	220	192	192	192	192	192	360	192	198	204	124	204	162	136	204	136	168	168	136	124	220	220	204	124	124
1-1-2-1	1st Stage	52	46	62	62	62	62	62	72	62	48	40	40	40	52	46	40	46	58	58	46	40	46	46	40	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	34	28	40	40	40	40	40	96	40	34	26	26	26	34	28	26	28	34	34	28	26	28	28	26	26	26
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	76	62	90	90	90	90	90	192	90	116	58	58	58	76	62	58	62	76	76	62	58	62	62	58	58	58
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	0	84	0	0	0	0	0	0	0	0	80	0	80	0	0	80	0	0	0	0	0	84	84	80	0	0
	Total	162	220	192	192	192	192	192	360	192	198	204	124	204	162	136	204	136	168	168	136	124	220	220	204	124	124
1-1-1-2	1st Stage	52	46	62	62	62	62	62	72	62	48	40	40	40	52	46	40	46	58	58	46	40	46	46	40	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	34	28	40	40	40	40	40	96	40	34	26	26	26	34	28	26	28	34	34	28	26	28	28	26	26	26
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	16	14	20	20	20	20	20	82	20	56	14	14	14	16	14	14	14	16	16	14	14	14	14	14	14	14
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A.2 (continued)

	4th Stage	60	132	70	70	70	70	70	110	70	60	124	44	124	60	48	124	48	60	60	48	44	132	132	124	44	44
	Total	162	220	192	192	192	192	192	360	192	198	204	124	204	162	136	204	136	168	168	136	124	220	220	204	124	124
1-1-1-1	1st Stage	52	46	62	62	62	62	62	72	62	48	40	40	40	52	46	40	46	58	58	46	40	46	46	40	40	40
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	34	28	40	40	40	40	40	96	40	34	26	26	26	34	28	26	28	34	34	28	26	28	28	26	26	26
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	16	14	20	20	20	20	20	82	20	56	14	14	14	16	14	14	14	16	16	14	14	14	14	14	14	14
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	60	48	70	70	70	70	70	110	70	60	44	44	44	60	48	44	48	60	60	48	44	48	48	44	44	44
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5th Stage	0	84	0	0	0	0	0	0	0	0	80	0	80	0	0	80	0	0	0	0	0	84	84	80	0	0
	Total	162	220	192	192	192	192	192	360	192	198	204	124	204	162	136	204	136	168	168	136	124	220	220	204	124	124

Table A.3 Heuristic Results of the 2nd Week of Case Study

Shop Design	Heuristic Method		Job Schedule	C_{max}
2-2-1	SPT&NEH Algorithm	FS1	J12-J15-J14-J04-J06-J09-J23-J19-J11-J16-J02-J20-J25	1304
		FS2	J21-J01-J10-J03-J05-J13-J24-J08-J07-J18-J17-J22-J26	
	LPT&NEH Algorithm	FS1	J11-J13-J16-J08-J24-J05-J07-J18-J01-J15-J20-J21-J26	1188
		FS2	J02-J04-J06-J22-J23-J10-J03-J09-J19-J14-J17-J12-J25	
	SPT	FS1	J12-J25-J15-J01-J18-J19-J04-J06-J10-J11-J16-J02-J23	1484
		FS2	J21-J26-J17-J20-J14-J03-J05-J07-J09-J13-J24-J22-J08	
	LPT	FS1	J08-J22-J16-J13-J10-J05-J03-J18-J01-J17-J26-J21-J12	1296
		FS2	J23-J02-J24-J11-J07-J09-J06-J04-J19-J14-J20-J15-J25	
2-1-2	SPT&NEH Algorithm	FS1	J23-J12-J02-J15-J14-J19-J16-J04-J11-J06-J09-J20-J25	1230
		FS2	J21-J22-J08-J17-J01-J24-J03-J05-J13-J10-J07-J18-J26	
	LPT&NEH Algorithm	FS1	J16-J23-J13-J08-J10-J04-J06-J09-J19-J15-J20-J21	1180
		FS2	J11-J02-J22-J24-J03-J05-J07-J18-J01-J14-J17-J12-J25-J26	
	SPT	FS1	J12-J25-J15-J20-J14-J19-J04-J06-J09-J11-J16-J02-J23	1390
		FS2	J21-J26-J17-J01-J18-J03-J05-J07-J10-J13-J24-J22-J08	
	LPT	FS1	J08-J22-J24-J13-J10-J07-J05-J03-J18-J01-J17-J26-J21	1266
		FS2	J23-J02-J16-J11-J09-J06-J04-J19-J14-J20-J15-J25-J12	
1-2-2	SPT&NEH Algorithm	FS1	J12-J15-J20-J14-J23-J04-J02-J06-J16-J09-J11-J19-J25	1392
		FS2	J21-J17-J08-J01-J03-J05-J22-J07-J13-J24-J10-J18-J26	
	LPT&NEH Algorithm	FS1	J23-J13-J16-J08-J10-J04-J06-J09-J19-J14-J17-J12-J25	1392
		FS2	J11-J02-J22-J24-J03-J05-J07-J18-J01-J15-J20-J21-J26	
	SPT	FS1	J12-J25-J26-J17-J15-J20-J14-J19-J04-J09-J11-J16	1596
		FS2	J21-J01-J18-J03-J05-J07-J10-J13-J24-J22-J06-J08-J02-J23	
	LPT	FS1	J08-J22-J24-J13-J06-J07-J04-J03-J18-J15-J17-J26-J12-J21	1392
		FS2	J23-J02-J16-J11-J09-J10-J05-J19-J14-J20-J01-J25	
2-1-1-1	SPT&NEH Algorithm	FS1	J12-J25-J26-J17-J01-J14-J19-J04-J22-J05-J02-J07-J13-J16-J10-J23-J09	1572
		FS2	J21-J15-J20-J18-J03-J06-J11-J24-J08	
	LPT&NEH Algorithm	FS1	J23-J13-J16-J06-J08-J24-J10-J04-J07-J18-J19-J14-J15-J17-J12-J21-J25	1550
		FS2	J11-J03-J02-J05-J22-J09-J01-J20-J26	
	SPT	FS1	J12-J25-J15-J20-J14-J19-J04-J06-J09-J11-J16-J02-J23	1736
		FS2	J21-J26-J17-J01-J18-J03-J05-J07-J10-J13-J24-J22-J08	
	LPT	FS1	J08-J22-J24-J13-J10-J07-J05-J03-J18-J01-J17-J26-J21	1688
		FS2	J23-J02-J16-J11-J09-J06-J04-J19-J14-J20-J15-J25-J12	
1-2-1-1	SPT&NEH	FS1	J12-J25-J26-J15-J17-J20-J01-J14-J18-J19-J23-J03-J22-J04-J02-J05-	1614

Table A.3 (continued)

	Algorithm		J24-J06-J16-J07-J11-J13-J09-J10	
		FS2	J21-J08	
	LPT&NEH Algorithm	FS1	J13-J23-J05-J07-J18-J14-J08-J24-J10-J03-J19-J15-J17-J20-J12-J21-J25-J26	1590
		FS2	J02-J04-J06-J09-J22-J01-J11-J16	
	SPT	FS1	J12-J25-J15-J20-J14-J19-J04-J06-J09-J11-J16-J02-J23	1700
		FS2	J21-J26-J17-J01-J18-J03-J05-J07-J10-J13-J24-J22-J08	
	LPT	FS1	J08-J22-J24-J13-J10-J07-J05-J03-J18-J01-J17-J26-J21	1736
		FS2	J23-J02-J16-J11-J09-J06-J04-J19-J14-J20-J15-J25-J12	
1-1-2-1	SPT&NEH Algorithm	FS1	J12-J15-J14-J19-J23-J04-J06-J02-J09-J16-J20-J11-J25	1404
		FS2	J21-J08-J01-J18-J03-J05-J10-J24-J22-J07-J13-J17-J26	
	LPT&NEH Algorithm	FS1	J13-J23-J08-J05-J24-J04-J06-J09-J19-J14-J17-J12-J25	1392
		FS2	J02-J11-J22-J16-J10-J03-J07-J18-J01-J15-J20-J21-J26	
	SPT	FS1	J12-J25-J26-J17-J15-J20-J14-J19-J04-J09-J11-J16	1650
		FS2	J21-J01-J18-J03-J05-J07-J10-J13-J24-J22-J06-J08-J02-J23	
	LPT	FS1	J08-J22-J24-J13-J06-J07-J04-J03-J18-J15-J17-J26-J12-J21	1392
		FS2	J23-J02-J16-J11-J09-J10-J05-J19-J14-J20-J01-J25	
1-1-1-2	SPT&NEH Algorithm	FS1	J12-J15-J20-J14-J23-J04-J02-J06-J11-J09-J16-J19-J25	1410
		FS2	J21-J17-J08-J01-J03-J05-J22-J07-J24-J10-J13-J18-J26	
	LPT&NEH Algorithm	FS1	J22-J11-J16-J08-J10-J04-J06-J09-J19-J14-J17-J12-J25	1392
		FS2	J02-J23-J13-J24-J03-J05-J07-J18-J01-J15-J20-J21-J26	
	SPT	FS1	J12-J25-J15-J01-J18-J19-J04-J06-J10-J11-J16-J02-J23	1596
		FS2	J21-J26-J17-J20-J14-J03-J05-J07-J09-J13-J24-J22-J08	
	LPT	FS1	J08-J22-J16-J13-J10-J05-J03-J18-J01-J17-J26-J21-J12	1392
		FS2	J23-J02-J24-J11-J07-J09-J06-J04-J19-J14-J20-J15-J25	
1-1-1-1-1	SPT&NEH Algorithm	FS1	J12-J25-J26-J15-J17-J20-J01-J14-J18-J19-J23-J03-J22-J04-J02-J05-J24-J06-J16-J07-J11-J13-J09-J10-J08	1596
		FS2	J21	
	LPT&NEH Algorithm	FS1	J16-J21-J22-J23-J09-J14-J11-J10-J05-J13-J24-J03-J04-J06-J07-J18-J19-J01-J15-J17-J20-J12-J25-J26-J08	1596
		FS2	J02	
	SPT	FS1	J12-J25-J15-J20-J14-J19-J04-J06-J09-J11-J16-J02-J23	1736
		FS2	J21-J26-J17-J01-J18-J03-J05-J07-J10-J13-J24-J22-J08	
	LPT	FS1	J08-J22-J24-J13-J10-J07-J05-J03-J18-J01-J17-J26-J21	1820
		FS2	J23-J02-J16-J11-J09-J06-J04-J19-J14-J20-J15-J25-J12	

Table A.4 Job List of the 3rd Week of Case Study

JOB NO	MODEL NAME	QUANTITY
J1	OSC 50 V	1
J2	OSC 25 TD	1
J3	OSC 60	1
J4	OSC 10 VT	1
J5	OSC 270D	1
J6	OSC 40	1
J7	OSC 75 U	1
J8	OSC 20 T-1	1
J9	OSC 7	1
J10	OSC 10 T	1
J11	OSC 40 VTD	1
J12	OSC 100 D	1
J13	OSC 125 U	1
J14	OSC 40 D	1
J15	OSC 7 T	1
J16	OSC 7 TD	1
J17	OSC 20 VT	1
J18	OSC 30	1
J19	OSC 25	1
J20	OSC 30 VTD	1
J21	OSC 75 V	1
J22	OSC 30 V	1
J23	OSC 340 U	1

Table A.5 Processing Times of Jobs for the 3rd Week Production Scheduling of Case Study

Shop Design	Processing Times (min)	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23
2-2-1	1st Stage	102	86	102	66	224	92	96	74	66	66	92	106	106	82	66	66	74	86	86	86	110	86	272
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	90	76	90	58	236	76	146	62	58	58	76	166	166	116	58	58	62	76	76	76	146	76	254
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	0	96	0	30	0	0	0	30	0	30	96	0	0	0	30	80	30	0	0	96	0	0	0
	Total	192	258	192	154	460	168	242	166	124	154	264	272	272	198	154	204	166	162	162	258	256	162	526
2-1-2	1st Stage	102	86	102	66	224	92	96	74	66	66	92	106	106	82	66	66	74	86	86	86	110	86	272
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	20	16	20	14	96	16	66	14	14	14	16	76	76	56	14	14	14	16	16	16	66	16	104
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	70	156	70	74	140	60	80	78	44	74	156	90	90	60	74	124	78	60	60	156	80	60	150
	Total	192	258	192	154	460	168	242	166	124	154	264	272	272	198	154	204	166	162	162	258	256	162	526
1-2-2	1st Stage	62	52	62	40	78	58	56	46	40	40	58	62	62	48	40	40	46	52	52	52	70	52	86
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	60	50	60	40	242	50	106	42	40	40	50	120	120	90	40	40	42	50	50	50	106	50	290
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	70	156	70	74	140	60	80	78	44	74	156	90	90	60	74	124	78	60	60	156	80	60	150
	Total	192	258	192	154	460	168	242	166	124	154	264	272	272	198	154	204	166	162	162	258	256	162	526
2-1-1-1	1st Stage	102	86	102	66	224	92	96	74	66	66	92	106	106	82	66	66	74	86	86	86	110	86	272
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	20	16	20	14	96	16	66	14	14	14	16	76	76	56	14	14	14	16	16	16	66	16	104
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	70	60	70	44	140	60	80	48	44	44	60	90	90	60	44	44	48	60	60	60	80	60	150

Table A.5 (continued)

	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	0	96	0	30	0	0	0	30	0	30	96	0	0	0	30	80	30	0	0	96	0	0	0
	Total	192	258	192	154	460	168	242	166	124	154	264	272	272	198	154	204	166	162	162	258	256	162	526
	1st Stage	62	52	62	40	78	58	56	46	40	40	58	62	62	48	40	40	46	52	52	52	70	52	86
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	60	50	60	40	242	50	106	42	40	40	50	120	120	90	40	40	42	50	50	50	106	50	290
1-2-1-1	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	70	60	70	44	140	60	80	48	44	44	60	90	90	60	44	44	48	60	60	60	80	60	150
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	0	96	0	30	0	0	0	30	0	30	96	0	0	0	30	80	30	0	0	96	0	0	0
	Total	192	258	192	154	460	168	242	166	124	154	264	272	272	198	154	204	166	162	162	258	256	162	526
	1st Stage	62	52	62	40	78	58	56	46	40	40	58	62	62	48	40	40	46	52	52	52	70	52	86
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	40	34	40	26	146	34	40	28	26	26	34	44	44	34	26	26	28	34	34	34	40	34	186
1-1-2-1	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	90	76	90	58	236	76	146	62	58	58	76	166	166	116	58	58	62	76	76	76	146	76	254
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	0	96	0	30	0	0	0	30	0	30	96	0	0	0	30	80	30	0	0	96	0	0	0
	Total	192	258	192	154	460	168	242	166	124	154	264	272	272	198	154	204	166	162	162	258	256	162	526
	1st Stage	62	52	62	40	78	58	56	46	40	40	58	62	62	48	40	40	46	52	52	52	70	52	86
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	40	34	40	26	146	34	40	28	26	26	34	44	44	34	26	26	28	34	34	34	40	34	186
1-1-1-2	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	20	16	20	14	96	16	66	14	14	14	16	76	76	56	14	14	14	16	16	16	66	16	104
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	70	156	70	74	140	60	80	78	44	74	156	90	90	60	74	124	78	60	60	156	80	60	150

Table A.5 (continued)

	Total	192	258	192	154	460	168	242	166	124	154	264	272	272	198	154	204	166	162	162	258	256	162	526
1-1-1-1-1	1st Stage	62	52	62	40	78	58	56	46	40	40	58	62	62	48	40	40	46	52	52	52	70	52	86
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	40	34	40	26	146	34	40	28	26	26	34	44	44	34	26	26	28	34	34	34	40	34	186
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	20	16	20	14	96	16	66	14	14	14	16	76	76	56	14	14	14	16	16	16	66	16	104
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	70	60	70	44	140	60	80	48	44	44	60	90	90	60	44	44	48	60	60	60	80	60	150
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5th Stage	0	96	0	30	0	0	0	30	0	30	96	0	0	0	30	80	30	0	0	96	0	0	0
	Total	192	258	192	154	460	168	242	166	124	154	264	272	272	198	154	204	166	162	162	258	256	162	526

Table A.6 Heuristic Results of the 3rd Week of Case Study

Shop Design	Heuristic Method		Job Schedule	C_{max}
2-2-1	SPT&NEH Algorithm	FS1	J14-J20-J12-J21-J05-J18-J01-J22-J10-J17-J09	1406
		FS2	J04-J07-J15-J02-J08-J19-J13-J23-J16-J11-J03-J06	
	LPT&NEH Algorithm	FS1	J13-J02-J21-J16-J23-J01-J17-J06-J04-J15-J19	1342
		FS2	J14-J12-J11-J20-J07-J05-J08-J03-J18-J10-J22-J09	
	SPT	FS1	J09-J10-J18-J22-J17-J01-J14-J07-J02-J11-J13-J23	1468
		FS2	J04-J15-J19-J08-J06-J03-J16-J21-J20-J12-J05	
	LPT	FS1	J23-J13-J11-J02-J07-J14-J01-J17-J22-J18-J10-J09	1526
		FS2	J05-J12-J20-J21-J16-J03-J06-J08-J19-J15-J04	
2-1-2	SPT&NEH Algorithm	FS1	J10-J11-J13-J18-J17-J02-J07-J23-J14-J01-J22-J09	1366
		FS2	J04-J15-J08-J19-J16-J20-J05-J12-J21-J03-J06	
	LPT&NEH Algorithm	FS1	J04-J13-J17-J02-J16-J08-J23-J09-J21-J03-J22	1266
		FS2	J11-J15-J20-J05-J12-J07-J06-J01-J10-J14-J18-J19	
	SPT	FS1	J09-J15-J19-J18-J17-J01-J14-J07-J02-J11-J23	1468
		FS2	J04-J10-J22-J08-J06-J03-J16-J21-J20-J12-J13-J05	
	LPT	FS1	J23-J13-J11-J02-J16-J06-J01-J19-J22-J18-J10	1504
		FS2	J05-J12-J20-J21-J07-J03-J14-J08-J17-J15-J04-J09	
1-2-2	SPT&NEH Algorithm	FS1	J05-J10-J18-J08-J06-J16-J20-J12-J21-J03-J09-J19	1502
		FS2	J23-J04-J15-J22-J17-J11-J07-J02-J13-J01-J14	
	LPT&NEH Algorithm	FS1	J11-J20-J16-J23-J18-J12-J03-J08-J19-J04-J15-J09	1402
		FS2	J02-J05-J17-J13-J21-J07-J01-J14-J10-J22-J06	
	SPT	FS1	J09-J10-J18-J22-J17-J06-J03-J01-J14-J07-J02-J11-J13	1694
		FS2	J04-J15-J19-J08-J16-J21-J20-J12-J05-J23	
	LPT	FS1	J23-J13-J11-J20-J14-J01-J17-J22-J15-J10-J09	1454
		FS2	J05-J12-J02-J07-J21-J16-J03-J06-J08-J19-J18-J04	
2-1-1-1	SPT&NEH Algorithm	FS1	J04-J15-J09-J19-J08-J06-J16-J20-J11-J07-J02-J23-J01-J14-J13	1646
		FS2	J10-J18-J22-J17-J03-J21-J12-J05	
	LPT&NEH Algorithm	FS1	J16-J01-J06-J12-J11-J20-J23-J08-J17-J21-J18-J19-J04-J15-J22-J09	1646
		FS2	J03-J07-J14-J05-J13-J02-J10	
	SPT	FS1	J09-J10-J18-J22-J17-J01-J14-J07-J02-J11-J13-J23	1646
		FS2	J04-J15-J19-J08-J06-J03-J16-J21-J20-J12-J05	
	LPT	FS1	J23-J13-J11-J02-J07-J14-J01-J17-J22-J18-J10-J09	1886
		FS2	J05-J12-J20-J21-J16-J03-J06-J08-J19-J15-J04	
1-2-1-1	SPT&NEH Algorithm	FS1	J04-J15-J09-J08-J17-J18-J19-J22-J13-J06-J01-J16-J07-J11-J03-J02-J20-J21-J14	1772
		FS2	J10-J12-J05-J23	

Table A.6 (continued)

	LPT&NEH Algorithm	FS1	J23-J13-J02-J20-J21-J16-J07-J14-J01-J03-J08-J17-J06-J18-J19-J04-J15-J22-J09-J10	1834
		FS2	J12-J11-J05	
	SPT	FS1	J09-J10-J18-J22-J17-J06-J03-J01-J14-J07-J02-J11-J13	1700
		FS2	J04-J15-J19-J08-J16-J21-J20-J12-J05-J23	
	LPT	FS1	J23-J13-J11-J20-J14-J01-J03-J22-J15-J10-J09-J04	1942
		FS2	J05-J12-J02-J07-J21-J16-J17-J06-J08-J19-J18	
1-1-2-1	SPT&NEH Algorithm	FS1	J23-J18-J07-J22-J13-J04-J11-J14-J02-J01-J17-J09	1552
		FS2	J12-J05-J19-J15-J21-J06-J20-J03-J16-J08-J10	
	LPT&NEH Algorithm	FS1	J23-J11-J02-J12-J07-J14-J08-J06-J18-J22-J09	1504
		FS2	J13-J05-J20-J16-J21-J01-J17-J03-J04-J15-J19-J10	
	SPT	FS1	J09-J15-J19-J18-J17-J01-J14-J07-J02-J11-J23	1754
		FS2	J04-J10-J22-J08-J06-J03-J16-J21-J20-J12-J13-J05	
	LPT	FS1	J23-J13-J11-J02-J16-J06-J14-J01-J15-J22-J18-J10	1542
		FS2	J05-J12-J20-J21-J07-J03-J08-J17-J19-J04-J09	
1-1-1-2	SPT&NEH Algorithm	FS1	J05-J10-J18-J08-J06-J16-J20-J12-J21-J03-J09-J19	1540
		FS2	J04-J23-J15-J22-J17-J11-J14-J02-J13-J07-J01	
	LPT&NEH Algorithm	FS1	J20-J23-J14-J11-J12-J03-J08-J07-J18-J22-J10-J09	1402
		FS2	J16-J02-J05-J13-J06-J21-J01-J17-J19-J04-J15	
	SPT	FS1	J09-J10-J18-J22-J17-J06-J03-J01-J14-J07-J02-J11-J13	1754
		FS2	J04-J15-J19-J08-J16-J21-J20-J12-J05-J23	
	LPT	FS1	J23-J13-J11-J20-J14-J01-J17-J22-J15-J10-J09	1532
		FS2	J05-J12-J02-J07-J21-J16-J03-J06-J08-J19-J18-J04	
1-1-1-1-1	SPT&NEH Algorithm	FS1	J10-J15-J09-J08-J17-J16-J18-J19-J22-J06-J01-J03-J11-J14-J02-J20-J07-J21-J12-J13-J05-J23	1754
		J21	J04	
	LPT&NEH Algorithm	FS1	J22-J09-J12-J13-J16-J01-J18-J10-J11-J21-J23-J08-J17-J06	1726
		FS2	J02-J07-J14-J03-J19-J04-J15-J05-J20	
	SPT	FS1	J09-J10-J18-J22-J17-J01-J14-J07-J02-J11-J13-J23	1754
		FS2	J04-J15-J19-J08-J06-J03-J16-J21-J20-J12-J05	
	LPT	FS1	J23-J13-J11-J02-J07-J14-J01-J17-J22-J18-J10-J09	1942
		FS2	J05-J12-J20-J21-J16-J03-J06-J08-J19-J15-J04	

Table A.7 Job List of the 4th Week of Case Study

JOB NO	MODEL NAME	QUANTITY
J1	OSC 180 U	1
J2	OSC 25 VTD	1
J3	OSC 40	1
J4	OSC 60 D	1
J5	OSC 340 D	1
J6	OSC 125 V	1
J7	OSC 40 T	1
J8	OSC 30 D	1
J9	OSC 75 D	1
J10	OSC 75 U	1
J11	OSC 270 U	1
J12	OSC 100 V	1
J13	OSC 50 U	1
J14	OSC 30 VTD	1
J15	OSC 30 TD	1
J16	OSC 40 TD	1
J17	OSC 40 VTD	1
J18	OSC 150 D	1
J19	OSC 150 D	1

Table A.8 Processing Times of Jobs for the 4th Week Production Scheduling of Case Study

Shop Design	Processing Times (min)	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19
2-2-1	1st Stage	168	86	92	92	272	114	92	80	96	96	224	114	92	86	86	92	92	168	168
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	192	76	76	134	254	166	76	116	146	146	236	166	134	76	76	76	76	192	192
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	0	96	0	0	0	0	36	0	0	0	0	0	0	96	96	96	96	0	0
	Total	360	258	168	226	526	280	204	196	242	242	460	280	226	258	258	264	264	360	360
2-1-2	1st Stage	168	86	92	92	272	114	92	80	96	96	224	114	92	86	86	92	92	168	168
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	82	16	16	64	104	76	16	56	66	66	96	76	64	16	16	16	16	82	82
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	110	156	60	70	150	90	96	60	80	80	140	90	70	156	156	156	156	110	110
	Total	360	258	168	226	526	280	204	196	242	242	460	280	226	258	258	264	264	360	360
1-2-2	1st Stage	72	52	58	52	86	70	58	46	56	56	78	70	52	52	52	58	58	72	72
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	178	50	50	104	290	120	50	90	106	106	242	120	104	50	50	50	50	178	178
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	110	156	60	70	150	90	96	60	80	80	140	90	70	156	156	156	156	110	110
	Total	360	258	168	226	526	280	204	196	242	242	460	280	226	258	258	264	264	360	360
2-1-1-1	1st Stage	168	86	92	92	272	114	92	80	96	96	224	114	92	86	86	92	92	168	168
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	82	16	16	64	104	76	16	56	66	66	96	76	64	16	16	16	16	82	82
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	110	60	60	70	150	90	60	60	80	80	140	90	70	60	60	60	60	110	110

Table A.8 (continued)

	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	0	96	0	0	0	0	36	0	0	0	0	0	0	96	96	96	96	0	0
	Total	360	258	168	226	526	280	204	196	242	242	460	280	226	258	258	264	264	360	360
1-2-1-1	1st Stage	72	52	58	52	86	70	58	46	56	56	78	70	52	52	52	58	58	72	72
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	178	50	50	104	290	120	50	90	106	106	242	120	104	50	50	50	50	178	178
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	110	60	60	70	150	90	60	60	80	80	140	90	70	60	60	60	60	110	110
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	0	96	0	0	0	0	36	0	0	0	0	0	0	96	96	96	96	0	0
	Total	360	258	168	226	526	280	204	196	242	242	460	280	226	258	258	264	264	360	360
1-1-2-1	1st Stage	72	52	58	52	86	70	58	46	56	56	78	70	52	52	52	58	58	72	72
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	96	34	34	40	186	44	34	34	40	40	146	44	40	34	34	34	34	96	96
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	192	76	76	134	254	166	76	116	146	146	236	166	134	76	76	76	76	192	192
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	0	96	0	0	0	0	36	0	0	0	0	0	0	96	96	96	96	0	0
	Total	360	258	168	226	526	280	204	196	242	242	460	280	226	258	258	264	264	360	360
1-1-1-2	1st Stage	72	52	58	52	86	70	58	46	56	56	78	70	52	52	52	58	58	72	72
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	96	34	34	40	186	44	34	34	40	40	146	44	40	34	34	34	34	96	96
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	82	16	16	64	104	76	16	56	66	66	96	76	64	16	16	16	16	82	82
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	110	156	60	70	150	90	96	60	80	80	140	90	70	156	156	156	156	110	110

Table A.8 (continued)

	Total	360	258	168	226	526	280	204	196	242	242	460	280	226	258	258	264	264	360	360
1-1-1-1-1	1st Stage	72	52	58	52	86	70	58	46	56	56	78	70	52	52	52	58	58	72	72
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2nd Stage	96	34	34	40	186	44	34	34	40	40	146	44	40	34	34	34	34	96	96
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3rd Stage	82	16	16	64	104	76	16	56	66	66	96	76	64	16	16	16	16	82	82
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4th Stage	110	60	60	70	150	90	60	60	80	80	140	90	70	60	60	60	60	110	110
	Buffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5th Stage	0	96	0	0	0	0	36	0	0	0	0	0	0	96	96	96	96	0	0
	Total	360	258	168	226	526	280	204	196	242	242	460	280	226	258	258	264	264	360	360

Table A.9 Heuristic Results of the 4th Week of Case Study

Shop Design	Heuristic Method		Job Schedule	C_{max}
2-2-1	SPT&NEH Algorithm	FS1	J13-J07-J16-J06-J03-J14-J10-J01-J19-J05	1526
		FS2	J17-J08-J15-J04-J02-J09-J12-J18-J11	
	LPT&NEH Algorithm	FS1	J04-J14-J09-J07-J18-J17-J06-J16-J05-J03	1434
		FS2	J08-J10-J12-J01-J15-J19-J02-J11-J13	
	SPT	FS1	J03-J07-J13-J10-J14-J16-J06-J01-J19-J05	1574
		FS2	J08-J04-J09-J02-J15-J17-J12-J18-J11	
	LPT	FS1	J05-J19-J01-J06-J16-J14-J10-J13-J07-J03	1690
		FS2	J11-J18-J12-J17-J15-J02-J09-J04-J08	
2-1-2	SPT&NEH Algorithm	FS1	J15-J17-J04-J02-J09-J12-J11-J18-J03	1428
		FS2	J14-J07-J08-J05-J10-J16-J01-J13-J19-J06	
	LPT&NEH Algorithm	FS1	J14-J05-J06-J17-J18-J09-J12-J13-J08	1400
		FS2	J19-J16-J02-J15-J11-J01-J10-J07-J04-J03	
	SPT	FS1	J03-J07-J04-J09-J14-J16-J12-J01-J19-J05	1526
		FS2	J08-J13-J10-J02-J15-J17-J06-J18-J11	
	LPT	FS1	J05-J19-J12-J06-J15-J14-J09-J10-J13-J03	1634
		FS2	J11-J18-J01-J17-J16-J02-J04-J07-J08	
1-2-2	SPT&NEH Algorithm	FS1	J14-J05-J01-J16-J19-J06-J10-J13-J07-J03	1422
		FS2	J02-J11-J15-J18-J09-J04-J17-J12-J08	
	LPT&NEH Algorithm	FS1	J02-J15-J19-J16-J05-J01-J10-J13-J08	1422
		FS2	J14-J11-J17-J09-J18-J06-J12-J07-J03-J04	
	SPT	FS1	J03-J07-J04-J09-J14-J16-J12-J06-J01-J05	1658
		FS2	J08-J13-J10-J02-J15-J17-J18-J11-J19	
	LPT	FS1	J05-J18-J01-J06-J17-J16-J14-J10-J13	1560
		FS2	J11-J19-J12-J15-J02-J09-J04-J07-J08-J03	
2-1-1-1	SPT&NEH Algorithm	FS1	J03-J04-J14-J15-J16-J09-J13-J19-J06-J12-J05-J18	1682
		FS2	J02-J07-J08-J17-J10-J01-J11	
	LPT&NEH Algorithm	FS1	J17-J05-J18-J19-J14-J15-J06-J12-J09-J10-J07-J04-J08-J03	1770
		FS2	J01-J16-J02-J11-J13	
	SPT	FS1	J03-J07-J04-J09-J14-J16-J12-J01-J19-J05	1716
		FS2	J08-J13-J10-J02-J15-J17-J06-J18-J11	
	LPT	FS1	J05-J19-J12-J06-J15-J14-J09-J10-J13-J03	1900
		FS2	J11-J18-J01-J17-J16-J02-J04-J07-J08	
1-2-1-1	SPT&NEH Algorithm	FS1	J07-J14-J03-J16-J04-J10-J06-J12-J18-J05	1690
		FS2	J02-J15-J08-J17-J13-J09-J01-J19-J11	
	LPT&NEH	FS1	J12-J01-J19-J17-J14-J15-J05-J09-J10-J04-J07-J13-J08-J03	1770

Table A.9 (continued)

	Algorithm	FS2	J06-J16-J02-J11-J18	
	SPT	FS1	J03-J07-J04-J09-J14-J16-J12-J06-J01-J05	1736
		FS2	J08-J13-J10-J02-J15-J17-J18-J11-J19	
	LPT	FS1	J05-J18-J01-J06-J17-J16-J14-J10-J13	1966
		FS2	J11-J19-J12-J15-J02-J09-J04-J07-J08-J03	
1-1-2-1	SPT&NEH Algorithm	FS1	J04-J18-J07-J17-J12-J11-J03-J14-J16-J09	1640
		FS2	J08-J15-J13-J02-J10-J06-J01-J19-J05	
	LPT&NEH Algorithm	FS1	J12-J09-J04-J05-J14-J07-J01-J17-J03-J19	1556
		FS2	J15-J10-J13-J06-J11-J16-J02-J18-J08	
	SPT	FS1	J03-J07-J13-J14 -J06-J01-J19-J05-J11	1736
		FS2	J08-J04-J09-J10-J02-J15-J16-J17-J12-J18	
	LPT	FS1	J05-J18-J12-J06-J16-J15-J10-J13-J04-J03	1722
		FS2	J11-J19-J01-J17-J14-J02-J09-J07-J08	
1-1-1-2	SPT&NEH Algorithm	FS1	J05-J16-J01-J13-J14-J19-J06-J10-J07-J03	1510
		FS2	J02-J17-J18-J04-J15-J11-J12-J09-J08	
	LPT&NEH Algorithm	FS1	J19-J12-J17-J14-J05-J01-J09-J07-J04-J03	1478
		FS2	J15-J11-J18-J13-J16-J02-J06-J10-J08	
	SPT	FS1	J03-J07-J04-J09-J14-J16-J12-J01-J19-J05	1736
		FS2	J08-J13-J10-J02-J15-J17-J06-J18-J11	
	LPT	FS1	J05-J19-J12-J06-J15-J14-J09-J10-J13-J03	1648
		FS2	J11-J18-J01-J17-J16-J02-J04-J07-J08	
1-1-1-1-1	SPT&NEH Algorithm	FS1	J15-J03-J02-J16	1730
		J21	J07-J14-J08-J17-J04-J13-J09-J10-J06-J12-J01-J18-J19-J05-J11	
	LPT&NEH Algorithm	FS1	J16-J02-J15-J10-J13-J01-J19-J12-J05-J03	1704
		FS2	J17-J08-J14-J09-J04-J07-J18-J06-J11	
	SPT	FS1	J03-J07-J13-J10-J14-J16-J06-J01-J19-J05	1736
		FS2	J08-J04-J09-J02-J15-J17-J12-J18-J11	
	LPT	FS1	J05-J19-J01-J06-J16-J14-J10-J13-J07-J03	1966
		FS2	J11-J18-J12-J17-J15-J02-J09-J04-J08	