# SCHEDULING IN A THREE-STAGE DEDICATED HYBRID FLOWSHOP WITH A COMMON THIRD-STAGE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ÇANKAYA UNIVERSITY

BY

SERDAR SOYSAL

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

SEPTEMBER 2008

Title of the Thesis : **Scheduling in a Three-stage Dedicated Hybrid Flowshop with a Common Third-stage**

Submitted by **Serdar SOYSAL**

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

[illegible]

Prof. Dr. Özhan ULUATAM
Acting Director

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

[illegible]

Prof. Dr. Levent KANDİLLER
Head of Department

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

[illegible]

Asst. Prof. Dr. Ferda Can ÇETİNKAYA
Supervisor

**Examination Date** : 12.09.2008

**Examining Committee Members**

| | | |
|---|---|---|
| Prof. Dr. Levent KANDİLLER | (Çankaya Univ.) | [illegible] |
| Asst. Prof. Dr. Ferda Can ÇETİNKAYA | (Çankaya Univ.) | [illegible] |
| Prof. Dr. Meral AZİZOĞLU | (METU) | [illegible] |

# STATEMENT OF NON-PLAGIARISM

| | |
|---|---|
| Name, Last Name | : Serdar SOYSAL |
| Signature | : [illegible] |
| Date | : 12.09.2008 |

## ABSTRACT

SCHEDULING IN A THREE-STAGE DEDICATED HYBRID FLOWSHOP WITH A COMMON THIRD-STAGE

SOYSAL, Serdar
M.Sc., Department of Industrial Engineering
Supervisor: Asst. Prof. Dr. Ferda Can ÇETİNKAYA

September 2008, 118 pages

In this study, we consider a scheduling problem of a manufacturing environment in which there are two manufacturing flow lines, where the third stage of the first line and the second stage of the second line are common. Each stage in the first flow line has a single machine whereas the second flow line contains two identical parallel machines in its first stage. Type-1 jobs are processed in the first flow line, whereas second flow line is dedicated to type-2 jobs. The last operation, of both types of jobs, must be processed on a common machine. The problem is to determine the sequence and schedule of all jobs at all stages of the two flow lines so that the makespan is minimized. We develop a mathematical model and a branch-and-bound algorithm with lower and upper bounding procedures to find optimal solution; we propose heuristic algorithms which provide good quality solutions at little computational effort when the computational effort to obtain an exact solution is prohibitive. The effectiveness of our solution approaches are demonstrated by computational analyses.

**Keywords:** Hybrid Flowshop Scheduling, Dedicated Machine, Mathematical Model, Branch-and-bound Algorithm.

# ÖZ

ÜÇÜNCÜ AŞAMASI ORTAK ÜÇ AŞAMALI TAHSİSLİ KARMA AKIŞ TİPİ

BİR ATÖLYEDE ÇİZELGELEME

SOYSAL, Serdar
Yüksek lisans, Endüstri Mühendisliği Anabilim Dalı
Tez Yöneticisi: Yrd.Doç. Dr. Ferda Can ÇETİNKAYA

Eylül 2008, 118 sayfa

Bu çalışmada, birincisinin üçüncü aşaması ile ikincisinin ikinci aşaması ortak olan iki imalat hattını barındıran bir imalat ortamının çizelgelenmesi problemi ele alınmıştır. Birinci imalat hattının her aşamasında tek makine mevcut iken, ikinci imalat hattının ilk aşamasında iki tane özdeş parallel makine yer almaktadır. Birinci tür işler ilk imalat hattında işlenirken ikinci imalat hattı ikinci tip işlere tahsis edilmiştir. Her iki tür işin son operasyonu ortak makinada işlenmek durumundadır. Problemimiz, bütün işlerin iki imalat hattının tüm aşamalarındaki sıra ve çizelgelerini belirlemek ve böylelikle başlangıç ve bitiş arasında geçen süreyi en aza indirmektir. Problemin optimal çözümünü bulmak için bir matematiksel model ile alt ve üst sınır işlemleriyle birlikte bir dal-sınır algoritması geliştirilmiştir; kesin çözümün elde edilemediği durumlarda, makul bir hesaplama uğraşıyla iyi çözümler sağlamak üzere sezgisel algoritmalar önerilmiştir. Çözüm yaklaşımlarımızın etkinliği sayısal analizlerle ispat edilmiştir.

**Anahtar Kelimeler:** Karma Akış Tipi Atölye Çizelgelemesi, Tahsisli Makine, Matematiksel Model, Dal-Sınır Algoritması.

## ACKNOWLEDGMENTS

I am deeply grateful my thesis supervisor Asst. Prof. Dr. Ferda Can ÇETİNKAYA for his precious and continual guidance and endless support. It was a great pleasure for me to have opportunity to study such a great researcher.

I would like to thank my jury members, Prof. Dr. Meral AZİZOĞLU and Prof. Dr. Levent KANDİLLER for their valuable contributions on this thesis. I am also thankful to Asst. Prof. Dr. Abdül Kadir GÖRÜR for his kind interest and supports. I am so lucky to have such a mentor and a friend.

I would like to express my deepest appreciation to my dear family. I am indebted to my mom Fatma ŞAHİN SOYSAL for her everlasting patience, empathy and encouragement. I am also grateful to my brother Tufan SOYSAL, my sister-in-law Belkız TAŞEL SOYSAL, my sister Eda SOYSAL ÖZHAN, and my brother-in-law Göktuğ ÖZHAN who listened my complaints and motivated me during this study. This study could not have been completed without their love.

I deem myself as fortunate being a part of ÇANKAYA-IE assistants and I would like to thank all my friends at the department, namely, Ender YILDIRIM, Engin TOPAN, İlter ÖNDER, İpek SEYRAN TOPAN, Miray Hanım ASLAN for their support to my study. I am especially appreciative to Ender YILDIRIM for his brotherhood.

I also would like to thank my dear friends Murat AKYAR, Sedef ELKER, Murat BAYAR, Gökhan ÖZHAN and Murat ŞAHİN for their invaluable friendship, help and moral support throughout this study.

Finally, thanks to TÜBİTAK (The Scientific and Technological Research Council of Turkey) for its support through a graduate study scholarship.

## TABLE OF CONTENTS

APPENDICES:

## LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

## INTRODUCTION

Scheduling is defined as the allocation of limited resources over time to perform a set of tasks. The resources and tasks may take many different forms. Machines, money, energy, processing units may represent resources, and operations of a process or part of a program may be considered as tasks.

In both manufacturing and service sectors, competition is getting harder. Time is a big pressure in such a market on the producers; customers have no tolerance for long lead times and products come out of fashion quickly. Because of that, efficient scheduling and sequencing have gained increasing importance in the enhancement of the productivity, utilization of the scarce resources, and profitability of the production lines. For high level performance, some production lines need special configuration and layout to process operations. For example, slower or overloaded manufacturing resources are duplicated in order to prevent any bottleneck or obstacle, hence balance the speed of production for all manufacturing stages as is in hybrid flowshop systems.

Hybrid flowshops are encountered in today's manufacturing environments quite often. It is a generalization of the flowshop and parallel machine environments, since it is a combination of these two systems where at least one stage of the hybrid flowshop environments contains more than one machine. Some examples of hybrid flowshop organizations from pharmaceutical industry and from semiconductor manufacturing industry are given in Artiba (1994) and in

Herrmann and Lee (1992), respectively. Besides, these organizations fit very well to glass container and glass manufacturing industries (He et al. (1996), Mei (1996), Paul (1979)), chemistry industry (Artiba and Riane (1998)), and cable manufacturing industry (Narasimhan and Panwalkar (1984)).

Many optimality criteria, like maximum lateness, total completion time, number of tardy jobs, total weighted tardiness are used as a measure of the performance in scheduling literature. The minimization of the makespan, which is the maximum completion time, is the most common one. Minimum makespan implies high utilization of the resources, gives room to early arrivals and early satisfaction of customer demand (Riane et al. (1998)).

Our study is concerned with scheduling two types of jobs on the three-stage dedicated flow lines where the third stage of the first line and the second stage of the second line are common. Each stage in the first flow line has a single machine whereas the second flow line contains two identical parallel machines in the first stage. Our aim is to minimize the maximum completion time of the jobs which is called as makespan.

We propose two heuristic algorithms for this *NP*-hard problem. Besides, we formulate the problem as a mixed integer program and develop a branch-and-bound algorithm with its lower and upper bounding procedures.

This thesis contains seven chapters that are organized as follows: In Chapter 2, we describe our problem and give its mathematical representation. In Chapter 3, we survey the related literature. The studies in the literature are classified according to the machine environment, as flowshop environment, identical parallel machine environment, hybrid flowshop environment with identical parallel machines and hybrid flowshop environment with dedicated machines. All these machine environments are reviewed for makespan objective. In Chapter 4, we present our branch-and-bound algorithm for the reversed problem. This algorithm gives us the exact solution. In Chapter 5, we propose two heuristic procedures based on the six different sequencing rules and a branch-and-bound based algorithm to be able to get a satisfactory; hopefully near optimal solution to our problem. Moreover, we discuss lower bounding procedures that are developed to evaluate the efficiency of the proposed heuristics. The sequencing rules are especially based on the well-known algorithm of Johnson (1954). Also, list scheduling procedures like Longest Processing Time (LPT) and Shortest Processing Time (SPT) are utilized in these heuristics. In Chapter 6, we discuss the results of our computational experiments to highlight the performance of our proposed heuristic, bounding procedures and the branch-and-bound algorithm. Finally, in Chapter 7, the study ends with some concluding remarks, together with some areas for future research.

# CHAPTER 2

## THREE-STAGE DEDICATED HYBRID FLOWSHOP WITH A COMMON THIRD-STAGE

In this chapter, we first define our problem together with its assumptions, and then present the associated mathematical model.

### 2.1 Problem Definition

The scheduling problem considered in this study can be stated as follows: Consider a manufacturing environment in which there are two flow lines having two and three stages. The third stage of the first line and the second stage of the second line are common. Each stage in the first flow line has a single machine whereas the second flow line contains two identical parallel machines in the first stage. There are two sets of $n = n_1 + n_2$ jobs, which are $J_1 = \{1, 2, ..., n_1\}$ and $J_2 = \{n_1 + 1, n_1 + 2, ..., n_1 + n_2\}$, where $J = J_1 \cup J_2$, and $n_1$ and $n_2$ are the number of type-1 and type-2 jobs, respectively. All jobs are simultaneously available at time zero, and each type-1 job has three operations, $O_j^1$, $O_j^2$, and $O_j^3$, with positive processing times $p_{j,1}$, $p_{j,2}$, and $p_{j,3}$; each type-2 job has two operations, $O_j^1$ and $O_j^3$, with positive processing times $p_{j,1}$ and $p_{j,3}$. The first and second operations of type-1 jobs must be processed at the first two stages of the first flow line, and the first operation of type-2 jobs must be processed by one of the identical machines at the first stage of the second flow line. The last operation, of all type-1 and type-2 jobs, must be processed on a common machine, which is utilized by the first and second flow lines at their last stage. Each machine at any stage of any flow line can handle no more than one job at a time. Also, each job can be processed by at most one machine at a time. Furthermore, we assume that preemption is not allowed; i.e., any operation once started must be completed without interruption. The problem, in such manufacturing environment, is to determine the sequence and schedule of all jobs at all stages of the two flow lines so that the makespan $C_{\max}$, which is the completion time of the job processed in the last position at the third stage, is minimized.

Scheduling problems can be denoted by a three-field notation $\alpha \mid \beta \mid \gamma$, where $\alpha$ specifies the machine environment, $\beta$ signifies the job and machine characteristics, and $\gamma$ denotes the objectives to be optimized. Following the standard three-field notation, we denote our problem of minimizing the makespan for a three-stage dedicated hybrid flowshop with a common third-stage by $F3 \mid k_1 = 3, k_2 = 1, k_3 = 1, T = 2 \mid C_{\max}$, where $k_i$ denotes number of machines at stage *i* and *T* denotes the number of job types, as illustrated by Figure 2.1.

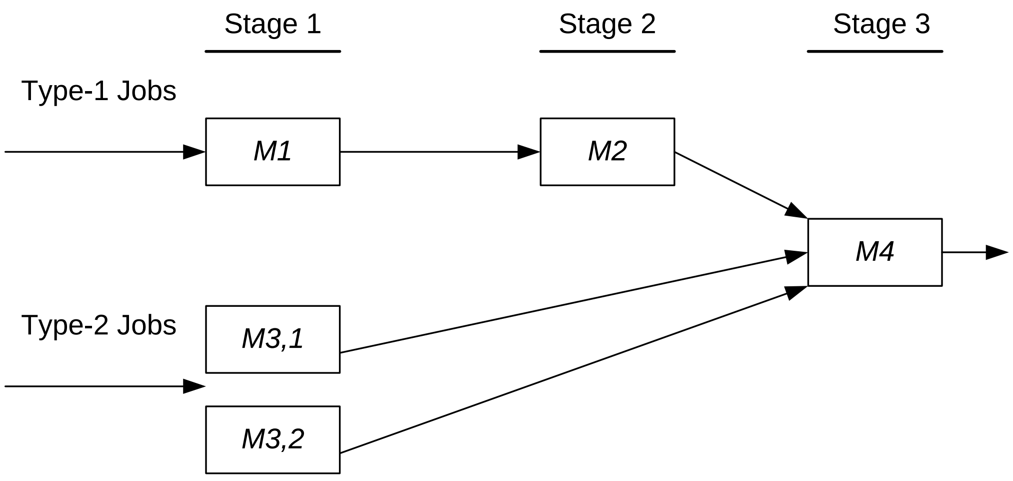

**Figure 2.1: A Three-stage Dedicated Hybrid Flowshop with a Common Third-stage Machine ($F3 \mid k_1 = 3, k_2 = 1, k_3 = 1, T = 2$)**

As an example of a practical application of this proposed model, consider the potato chips and pop corn production in the same workshop. In such environments, there are mainly two types of product families, which are potato products and corn products. Two separate flow lines are dedicated for each product family. Each product in a potato products family is processed at the first two stages of its associated flow line, where peeling and frying operations are performed, whereas each product in a corn products family is processed by one of the two identical machines at the first stage of its associated flow line, where frying operation is performed, and then the products are packed in bags at the last stage. Another application occurs in a manufacturing environment in which initial operations of two different product families are processed at two separate flow lines but each product family must go through a final quality control operation, which is to be carried out on a common testing machine.

In this thesis, we extend the work of Oğuz et al. (1997). They studied a similar problem for the two-stage flow lines, where each stage consists of one machine only, and proved that the two-stage flow line problem belongs to the class of *NP*-hard problems. For this reason, they proposed a heuristic algorithm and analyzed its worst-case error bound. They also derived a global lower bound to computationally test the performance of their proposed heuristic.

Observe that when all operations $O_j^3$ have zero processing time, the problem reduces to two separate problems: a classical two-stage flowshop problem which can be solved in $O(n \log n)$ time using the well-known algorithm of Johnson (1954) and a parallel machine problem with two identical machines which is proved to be *NP*-hard by Karp (1972). When operations $O_j^1$ and $O_j^3$ have zero processing time for type-2 jobs, then the resulting problem is one of scheduling $n_1$ jobs on three-stage flowshop to minimize the makespan, which is known to be *NP*-hard by Garey et al. (1976). Further, when operations $O_j^2$ have zero processing time for type-1 jobs, and one of the identical parallel machines at the second flow line is removed, the problem reduces to the dedicated two-stage flowshop scheduling with a common second-stage machine, which was proved to be strongly *NP*-hard by Oğuz et al. (1997). Thus, our problem is also strongly *NP*-hard.

## 2.2 Mathematical Model

The problem addressed in this research can also be expressed as a scheduling problem for a four-stage hybrid flowshop where the first two stages are for the first flow line, third stage is for the parallel machines of the second flow line and the fourth stage is the common stage. In this case, note that the processing times of type-1 jobs in the third stage and the processing times of type-2 jobs in the first two stages are all zero. The sequence of jobs on the machines and the completion time of the jobs at each stage define the decision variables. The indices, parameters and decision variables are as follows:

*Indices*

$i, r$ : index for jobs ($i, r = 1, ..., n_1, n_1 + 1, ..., n_1 + n_2$)

$j$ : index for stages ($j = 1,..,4$)

*Parameters*

$p_{i,j}$ : processing time of job $i$ at stage $j$

$M$ : a very large positive number

*Decision Variables*

$C_{i,j}$ : completion time of job $i$ at stage $j$

$C_{\max}$ : maximum completion time (makespan)

$$Y_i = \begin{cases} 1, & \text{if} \quad \text{job } i \text{ on stage 3 is asigned to machine 1} \\ 0, & \text{otherwise} \end{cases}$$

$$X_{i,r,j} = \begin{cases} 1, & \text{if} \quad \text{job } i \text{ precedes job } r \text{ on stage } j \\ 0, & \text{otherwise} \end{cases}$$

$Y_i$ and $X_{i,r,j}$ are binary variables. The rest of the variables are continuous that take integer values when the processing times are given as integers.

Given the above-defined variables and parameters, the scheduling problem of the three-stage dedicated hybrid flowshop with a common third stage can be formulated as follows:

Minimize $\quad C_{\max}$

Subject to

(1) Makespan cannot be smaller than the completion time of any job at the common stage.

$$C_{\max} \geq C_{i,4} \qquad \text{for } i = 1,\ldots,n_1 + n_2 \qquad (2.1)$$

(2) Completion time of any type-1 job at stages 1, 2 and 3 cannot be less than its completion time at the previous stage.

$$C_{i,j} \geq C_{i,j-1} + p_{i,j} \qquad \text{for } i = 1,\ldots,n_1; \quad j = 1,2,3 \qquad (2.2)$$

(3) Completion time of any type-2 job at parallel machine environment must be greater than or equal to the processing time of it in the same stage.

$$C_{i,3} \geq p_{i,3} \qquad \text{for } i = n_1 + 1,\ldots,n_1 + n_2 \qquad (2.3)$$

(4) Completion time of any job before the common stage cannot be less than its completion time at the previous stage.

$$C_{i,4} \geq C_{i,3} + p_{i,4} \qquad \text{for } i = 1,\ldots,n_1 + n_2 \qquad (2.4)$$

(5) It is necessary to have non-interference for every pair of jobs $i$ and $r$, either $i$ precedes $r$ or the other way around. Hence, the difference between the processing times of any two jobs at any stage must be such that they do not overlap.

$$M\ X_{i,r,j} + C_{i,j} - C_{r,j} \geq p_{i,j} \qquad \text{for } i = 1,\dots,n_1;\ j = 1,2;\ r = 1,\dots,n_1;\ i < r \tag{2.5}$$

$$M\ (1 - X_{i,r,j}) + C_{r,j} - C_{i,j} \geq p_{r,j} \qquad \text{for } i = 1,\dots,n_1;\ j = 1,2;\ r = 1,\dots,n_1;\ i < r \tag{2.6}$$

$$M\left(2 - Y_i - Y_r + X_{i,r,3}\right) + C_{i,3} - C_{r,3} \geq p_{i,3} \qquad \text{for } i = n_1+1,\dots,n_1+n_2;\ r = n_1+1,\dots,n_1+n_2;\ i < r \tag{2.7}$$

$$M\left(Y_i + Y_r + X_{i,r,3}\right) + C_{i,3} - C_{r,3} \geq p_{i,3} \qquad \text{for } i = n_1+1,\dots,n_1+n_2;\ r = n_1+1,\dots,n_1+n_2;\ i < r \tag{2.8}$$

$$M\left(3 - Y_r - Y_i - X_{i,r,3}\right) + C_{r,3} - C_{i,3} \geq p_{r,3} \qquad \text{for } i = n_1+1,\dots,n_1+n_2;\ r = n_1+1,\dots,n_1+n_2;\ i < r \tag{2.9}$$

$M\left(1+Y_r+Y_i-X_{i,r,3}\right)+C_{r,3}-C_{i,3} \geq p_{r,3}$ for $i=n_1+1,\ldots,n_1+n_2$;

$r=n_1+1,\ldots,n_1+n_2$;

$i<r$ (2.10)

$M\ X_{i,r,4}+C_{i,4}-C_{r,4} \geq p_{i,4}$ for $i=1,\ldots,n_1+n_2$;

$r=1,\ldots,n_1+n_2$;

$i<r$ (2.11)

$M\ (1-X_{i,r,4})+C_{r,4}-C_{i,4} \geq p_{r,4}$ for $i=1,\ldots,n_1+n_2$;

$r=1,\ldots,n_1+n_2$;

$i<r$ (2.12)

(6) Constraint sets (2.13) and (2.14) enforce the integrality for $Y_i$ and $X_{i,r,j}$.

$Y_i \in \{0,1\}$ for $i=n_1+1,\ldots,n_1+n_2$ (2.13)

$X_{i,r,j} \in \{0,1\}$ for $i=1,\ldots,n_1+n_2$;

$r=1,\ldots,n_1+n_2$;

$j=1,..,4$

$i<r$ (2.14)

(7) Completion time of any job at any stage cannot be negative.

$C_{i,j} \geq 0$ for $i=1,\ldots,n_1+n_2$;

$j=1,..,4$ (2.15)

The proposed heuristic algorithm in Section 5.1 can be used to derive an upper bound for the makespan. Instead of assuming the initial upper bound of infinity, the makespan value obtained from the heuristic algorithms can used as an upper bound. Moreover, the global lower bound value in Section 5.3 can also be utilized as a lower bound. Thus, the following constraint can be added to the above model,

$$LB \le C_{\max} \le UB \qquad (2.16)$$

where $UB = \min\left\{C_{\max}\left(\text{Rule } k\right)\right\};\ k \in \left\{A,...F\right\}$ and $LB = \max\limits_{i=1,...,4}\left\{LB_i\right\}$.

Number of constraints, binary variables, and continuous variables in our mathematical model can be computed as follows:

*Constraints*

| | |
|---|---|
| $n$ | constraints from equation (2.1) where $n = n_1 + n_2$ |
| $3n_1$ | constraints from equation (2.2) |
| $n_2$ | constraints from equation (2.3) |
| $n$ | constraints from equation (2.4) |
| $2\times\left(\frac{(n_1-1)(n_1)}{2}\right)$ | constraints from equation (2.5) |
| $2\times\left(\frac{(n_1-1)(n_1)}{2}\right)$ | constraints from equation (2.6) |
| $\left(\frac{(n_2-1)(n_2)}{2}\right)$ | constraints from equation (2.7) |
| $\left(\frac{(n_2-1)(n_2)}{2}\right)$ | constraints from equation (2.8) |

| | |
|---|---|
| $\left(\frac{(n_2-1)(n_2)}{2}\right)$ | constraints from equation (2.9) |
| $\left(\frac{(n_2-1)(n_2)}{2}\right)$ | constraints from equation (2.10) |
| $\left(\frac{(n-1)(n)}{2}\right)$ | constraints from equation (2.11) |
| $\left(\frac{(n-1)(n)}{2}\right)$ | constraints from equation (2.12) |
| $n_2$ | constraints from equation (2.13) |
| $4\times\left(\frac{(n-1)(n)}{2}\right)$ | constraints from equation (2.14) |
| $4n$ | constraints from equation (2.15) |
| 1 | constraint from equation (2.16) |

Number of constraints = $6n+3(n-1)n+3n_1+2(n_1-1)n_1+2(n_2+(n_2-1)n_2)+1$

$$= 3n^2+2n_1{}^2+2n_2{}^2+3n+n_1+1$$

*Binary Variables*

| | |
|---|---|
| $n_2$ | binary variables from $Y_i$ |
| $4\left(\frac{(n-1)(n)}{2}\right)$ | binary variables from $X_{i,r,j}$ |

Number of binary variables = $n_2+2(n-1)n$

*Continuous Variables*

| | |
|---|---|
| $4n$ | continuous variables from $C_{i,j}$ |
| $4\left(\frac{(n-1)(n)}{2}\right)$ | continuous variables from $C_{\max}$ |

Number of continuous variables = $4n + 2(n-1)n \; = \; 2n^2 + 2n$

# CHAPTER 3

## LITERATURE REVIEW

Hybrid flowshop systems are encountered in today’s manufacturing environment quite often. Having a wide area of applicability reveals the importance of the system and researchers have directed their studies on the hybrid flowshop scheduling problems. Hybrid flowshop environment is closely related with flowshop and parallel machine environments, since it is a skillful combination of these two items in such a manner that at least one stage of the hybrid flowshop environments contains more than one machine.

We survey the related literature according to the machine environment, as flowshop environment, identical parallel machine environment, hybrid flowshop environment with identical parallel machines and dedicated machine hybrid flowshop environment. All these machine environments are reviewed according to the objective function makespan.

### 3.1 Flowshop Scheduling Problem

“In the flowshop scheduling problem, we are given machines $M_1$, $M_2$,…, $M_m$, where $m \geq 2$ and a set $N = \{1, 2,\ldots, n\}$ of jobs. Each job has to be processed first on $M_1$, then on $M_2$, and so on, until it is processed on the last machine $M_m$” (Chen et al. (1996)).

### 3.1.1 Two-Machine Problem

The optimal solution for the makespan problem in the two-machine flowshop is presented in S. M. Johnson's (1954) famous paper, which is the pioneer work on these problems. In this paper, Johnson proposes the following optimizing algorithm.

**Johnson's Algorithm (1)**

**Step 1:** Find the minimum processing time among unscheduled jobs.

**Step 2:** If the minimum in Step 1 occurs on machine 1, place the associated job in the first available position in sequence (Ties may be broken arbitrarily), and go to Step 3; otherwise, place the associated job in the last available position in sequence (Ties may be broken arbitrarily).

**Step 3:** Remove the assigned job from the list of unscheduled jobs and return to Step 1 until all sequence positions are filled.

An alternative statement of this algorithm is provided by using a different perspective (Baker (1995), pp. 8.7-8.8).

**Johnson's Algorithm (2)**

**Step 1:** Partition the jobs into two sets, according to whether the first operation is shorter or longer than the second operation. In case of tie, the job can be placed into any set arbitrarily.

**Step 2:** Sequence the jobs with shorter first operations in nondecreasing order of their processing time on machine 1, and sequence the jobs with longer first operations in nonincreasing order of their processing time on machine 2.

**Step 3:** Arrange the two sequences in tandem to produce a full sequence for the solution.

### 3.1.2 Multi-Machine Problem

Johnson's Algorithm cannot be generalized for flowshops with more than two machines. However, there are some special polynomially solvable cases for flowshops with three machines. Burns and Rooker (1976, 1978) and Szwarc (1977) discuss these special cases. Smits and Baker (1981) design experiments to test these special cases on sample problems. They present special conditions that optimal solutions to three-machine flowshop problem can be found by a polynomial algorithm. Garey et al. (1976) study the same problem. They show that the $F_3 \| C_{\max}$ problem is *NP*-hard in strong sense.

Several optimizing algorithms have been proposed for scheduling problems with $n$ jobs on $m$ machines in a flowshop environment to minimize the makespan. In these algorithms, numerous assumptions are made. Dudek and Teuton (1964) provide all these assumptions as follows:

**A.** Assumptions regarding machines:

1. No machine may process more than one job at any given time and each job, once started, must be processed to completion.
2. A known, finite time, is required to perform each operation and the time intervals for processing are independent of the order in which the operations are performed.

**B.** Assumptions regarding jobs:

1. All jobs are known and are completely organized for processing before the period under consideration begins.
2. All jobs are considered equal in importance, i.e., there are no due dates.
3. Jobs are processed by the machines as soon as possible and in a common order.

**C.** Other:

1. The time required to transport jobs between machines may be considered negligible or as a part of the processing time on the preceding machine.
2. In-process inventory is allowable.

Palmer (1965) studies the $F_m \| C_{\max}$ problem and proposes a heuristic algorithm which is called as the slope order heuristic. The heuristic is based on the idea that give precedence to the jobs having the strongest tendency to progress from short times to long times in the sequence of processes. Campbell et al. (1970) show that optimal or near-optimal solutions can be produced to the $F_m \| C_{\max}$ problem by using the heuristic algorithm CDS. The algorithm constructs $m-1$ artificial two-machine problems from the original $m$-machine problem. Then, these artificial problems are solved by using Johnson's two-machine algorithm, and the best solution among them becomes the solution to the original problem. The authors compare their algorithm with Palmer's slope order heuristic by solving many problem instances. Their computational experiments show that computation time of Palmer's heuristic is less than the computation time of CDS. However, CDS is superior to the slope order heuristic in point of producing solutions with less average error percentage. Page (1961) studies the same problem. He proposes four heuristics which are mainly based on the sorting techniques. Two of these heuristics, individual exchange heuristic and group exchange heuristic, are improvement procedures. They attempt to find better solutions by using initial solutions. The other ones, merging and pairing heuristics, generate initial solutions. Computational experiments show the efficiency of these heuristic algorithms. Gupta (1971) develops a heuristic algorithm, functional heuristic algorithm, as an extension of Page's analogy of scheduling and sorting. His heuristic is capable to solve problems optimal or near optimal. Many experiments are conducted to compare the performance of the proposed heuristic with Palmer's slope order heuristic. The algorithms are compared in point of the two criteria. One of them is algorithms' effectiveness in finding better solution and the other one is computational times required to obtain the solution. Solutions obtained by this algorithm are better than the Palmer' slope order heuristic.

Dannenbring (1977) studies the same problem and explores neighborhood search techniques. He proposes three new heuristics which are rapid access procedure (RA), rapid access with close order search (RACS), and rapid access with extensive search (RAES). RA generates a good starting solution quickly and easily; RACS and RAES take this solution as initial and attempt to find better solutions. His computational experiments show the efficiency of his algorithms. Nawaz et al. (1983) propose a heuristic, NEH heuristic, based on the idea that jobs with more processing time on all the machines should be given priority and scheduled earlier than the ones with less total processing time on all the machines. Initially, first two jobs are scheduled in order to minimize makespan; then other jobs are inserted in the partial schedule one by one to find the best schedule at every step. And, finally a complete schedule is obtained. Computational experiments show that the NEH heuristic is capable to solve problems optimal or near optimal. Framinan et al. (2003) modify the NEH heuristic by using different initial orders based on the approaches in the literature. Jobs are not only sorted according to the sum of processing time; also sum of absolute differences of a job's processing time on every single machine with other jobs, or sum of absolute residuals (Stinson and Smith (1982)), etc are used to sort them. Among these results, the one with the smallest value is accepted as the solution of the proposed heuristic. Their computational studies show that for the makespan minimization problem original NEH heuristic is the best solution, but for different objective functions, better solutions are obtained with different initial orders. Koulamas (1998) proposes the HFC heuristic for the same problem. His heuristic is based on the Johnson's algorithm and has two phases. In the first part, a sequence is produced by extensive use of Johnson's algorithm; while in the second part, this sequence is used as an input and by allowing non-permutation schedules an improvement is expected. Suliman (2000) proposes a two-phase heuristic for the problem. In the first phase of the heuristic, well-known heuristics like CDS, Palmer's slope index, RA, are used to generate initial job sequence, and in the second phase, this initial sequence is improved by using a pair exchange mechanism. The performance of the heuristic is compared with the other heuristic in the literature, and computational experiments express the efficiency of the heuristic.

Taillard (1990) applies tabu search technique to the same problem. He addresses that good solutions are found for the randomly generated instances by using this technique; however, it needs great calculation times. Computational experiments demonstrate the efficiency of the algorithm. Ponnambalam et al. (2001) propose another metaheuristic algorithm, genetic algorithm, to solve the problem. Their computational experiments report the performance of the algorithm.

Wagner (1959) studies the same problem. He provides a mixed integer programming model to solve small size problem instances. Computational experiments show the performance of the model.

Ignall and Schrage (1965) apply the branch-and-bound technique to the three-machine flowshop problem to minimize the makespan. In order to use this technique, the problem is described as a tree which each node represents a partial solution. At the roof of this tree, there is only one node and this node has $n$ number of branches that equals to the number of jobs in the problem. One of these branches is chosen and the job that is represented by the node at the end of this branch is assigned to the first position in the sequence. So, in the next level, there are only $n-1$ branches to assign the second position. The tree structure is shaped in this manner until last job is assigned to the last position in the sequence. The authors propose a lower bound to be used in branching procedure. At the each level of the tree, the node with the smallest lower bound is selected and the tree is branched from this node. Their branch-and-bound algorithm is capable of solving problem instances with up to 10 jobs. Lomnicki (1965) studies the same problem and proposes an efficient lower bound for his branch-and-bound algorithm. His computational experiments show the efficiency of his algorithm.

Lomnicki, Ignall and Schrage obtain their lower bound by considering the total processing time on one machine. McMahon and Burton (1967) form a new lower bound that is based on the idea of determining makespan by the total processing time for a job rather than by the total processing time on one machine. The authors also propose some methods to improve the efficiency of their algorithm. By placing the dominant machine, which has the greater total processing time than the other machine, last the efficiency of the algorithm is increased. The advantage of the fact that scheduling problems are symmetrical with respect to time-reversal is taken in that point. They test the performance of their lower bound with respect to Lomnicki and Ignall and Schrage's lower bounds and report that it performs better than the others. Besides, their computational experiments show that placing dominant machine last increases the performance of the algorithm.

Ruiz and Maroto (2005) and Hejazi and Saghafian (2005) present a review of the previous studies. Exact methods, constructive and improvement heuristics are surveyed and their performances are pointed out.

## 3.2 Parallel Machines Scheduling Problem

In the classical parallel machines scheduling problem, there are $n$ jobs and $m$ machines in parallel. Furthermore, each job is processed on one of the *m* machines during a fixed processing time with keeping the job on the machine until completion.

Machines are considered as identical if they have the same speed. On the identical parallel machines, the processing time of each job are not affected by the machine processing it, each can be processed on any of the machines.

In the following subsection, identical parallel machines scheduling problems without preemption, keeping the job on machine until completion, and precedence among jobs are reviewed for the makespan.

### 3.2.1 Identical Parallel Machine Problem

Pioneer work on the complexity of the $P_m \,||\, C_{\max}$ problem is done by Karp (1972). He shows that the $P_2 \,||\, C_{\max}$ problem is *NP*-hard. Since there is no way to find optimization polynomial time algorithm for $P_m \,||\, C_{\max}$, some approximation algorithms are used to evaluate the problems' worst case and mean behavior. One of the most used approximation algorithm for scheduling problem is list scheduling. The main idea in list scheduling is forming an ordered list of processes by giving them priorities. Then, at each step the job with the highest priority in the list is assigned to the first available machine (Graham (1966)). One of the well-known list scheduling algorithms is longest processing time (LPT) algorithm which the jobs are arranged in non-increasing order of processing time. Graham (1969) shows that $C_{\max}$ value of the LPT rule is at most $\frac{4}{3}-\frac{1}{3m}$ times the optimal $C_{\max}$ value for the problems with two or more identical parallel machines. Coffman and Sethi (1976) suggest another absolute performance ratio for LPT rule. The new worst case error bound is $1+\frac{1}{k}-\frac{1}{km}$ where $k$ is the number of tasks on any machine whose last task terminates the schedule. They show that this error bound is tight when there are three or more machines. And, for the two and one machine cases, the worst case error bound is 1. Coffman et al. (1984) present how good the LPT algorithm is on the average for two machine environment under the assumptions that task processing times are independent samples and uniformly distributed on [0,1]. They prove that mean value of schedule length for the LPT algorithm, $E(C_{\max}^{LPT})$, is bounded with

$$\frac{n}{4}+\frac{1}{4(n+1)} \le E(C_{\max}^{LPT}) \le \frac{n}{4}+\frac{e}{2(n+1)}$$

where $e$ is the base of the natural algorithm and $n$ is the number of task. Frenk and Rinnooy Kan (1984) show that the absolute error between $C_{\max}$ value of LPT and $C_{\max}$ value of optimum (i.e., $C_{\max}^{LPT}-C_{\max}^{OPT}$) converges to zero almost surely as well as in expectation as number of tasks goes to infinity. They consider special cases of the uniform and exponential distributions to analyze the speed at which the absolute error converges to 0. In Frenk and Rinnooy Kan (1986), the authors extend and generalize the results. They prove that rate of convergences for almost sure and in expectation are $O\left(\log\left(\log n\right)/n\right)$ and $1/n$ respectively.

Coffmann et al. (1978) presents a new approximation algorithm, multifit algorithm, to get better performance guarantees. In this algorithm, it is tried to determine whether a schedule can be constructed that is consistent with the smallest feasible value of makespan (*M*) by using a heuristic procedure known as first-fit decreasing (FFD). This procedure operates by first sorting the tasks according to LPT, and then inserting each task one by one into the first machine so that it completes before or on *M*. Then, jobs are assigned to another machine. This is repeated until all jobs are scheduled. If any partial schedule exceeds on any machine, then the procedure fails. Friesen and Langston (1986) prove that $C_{\max}$ value of multifit algorithm is at most $\frac{72}{61}$ times the optimal $C_{\max}$ value. Although multifit produces tighter bounds than LPT, it requires more computational effort. An example is given that multifit produces worse makespan than LPT in (Baker (1995), pp. 7.6-7.7). Lee and Massey (1988) suggest combining LPT and multifit to improve performance. First a schedule is arranged by using LPT rule, then this schedule is used as an initial upper bound and multifit search is performed in the interval that is bounded on the upper side with this bound. Their computational experiments show that combined algorithm needs less searching them multifit alone. Haouari et al. (2006) present a review of the previous studies on bounds for the identical parallel machine scheduling problem and recommend new lower bounding strategies and heuristics for this scheduling problem. The lower bounds are based on the lifting procedure. Their optimization-based heuristic yields the solution by iteratively solving a subset-sum problem. The authors' computational experiments show the satisfactory performance of their lower bound strategy and algorithm. The algorithm performs well on wide range of instances.

Rothkopf (1966) studies the $P_m \| C_{\max}$ problem. He develops an exponential-time algorithm, based on a dynamic programming approach, which only solves small instances. Mokotoff (2002) studies the same problem and proposes an exact cutting plane algorithm built from identification of valid inequalities that apply to the subset of the solutions by a maximum value of the makespan. With these inequalities, constraints are generated for the algorithm and these constraints are added iteratively starting from the solution obtained by successive linear programming relaxation. Upper and lower bounds are also used in the algorithm to narrow the interval that will be searched. A simple lower bound is obtained by giving permission to preemption and an upper bound is obtained by using the most known procedures, i.e. LPT, multifit. His computational experiments show the efficiency of the algorithm to produce quality feasible solutions. Lee et al. (2006) propose simulated annealing heuristic algorithm, Min and Cheng (1999) develop genetic algorithm for minimizing makespan value on identical parallel machines. Computational experiments show the efficiency of these algorithms.

## 3.3 Hybrid Flowshop Scheduling Problem

A hybrid flowshop is a generalization of the flowshop and the parallel machine environments. It consists of a series of production stages in series instead of *m* machines in series. And, at least one of these stages has multiple machines in parallel. Each job is processed by one machine in each stage and it flows through one or more stage. Besides, all the assumptions for flowshop and parallel machine scheduling problems are valid for hybrid flowshop scheduling problems.

In the next two subsections, hybrid flowshop scheduling problems with identical parallel machines in at least one of the stages are surveyed according to the makespan.

### 3.3.1 Two-Stage Hybrid Flowshop

Arthanari and Ramamurthy (1971) study the two-stage hybrid flowshop problem, where there is only one machine at the second stage. They suggest several lower bounds and a branch-and-bound algorithm to minimize makespan. However, they test this exact algorithm only for small instances that are less than ten jobs. Gupta (1988) proves that the two-stage flowshop problem when there are identical multiple machines at each stage to minimize makespan is *NP*-complete. He develops a heuristic algorithm, which is based on the Johnson's rule, for the special case when there is one machine at the second stage. His computational experiments are limited with two machines at the first stage.

Sriskandarajah and Sethi (1989) study the performance of algorithms for minimizing makespan schedules. The authors show that for Johnson's algorithm applied to problems $F2 \mid k_1 = 1, k_2 = k = 2 \mid C_{\max}$ and $F2 \mid k_1 = 1, k_2 = k \geq 3 \mid C_{\max}$ with $C_{\max} \leq \sum_{j=1}^{n} p_{1j} + \max_j \{ p_{2j} \}$ where $n$ is the number of jobs, $p_{1j}$ and $p_{2j}$ are processing times of job j at stages 1 and 2, respectively; the best possible bound is $C_{\max} / C_{\max}^{optimum} \leq 2$. Also, for Johnson's algorithm applied to problems $F2 \mid k_1 = 1, k_2 = k \geq 3 \mid C_{\max}$ with $C_{\max} > \sum_{j=1}^{n} p_{1j} + \max_j \{ p_{2j} \}$, the bound is $C_{\max} / C_{\max}^{optimum} \leq 1 + \left(2 - \frac{1}{k}\right)\left(1 - \frac{1}{k}\right)$. Moreover, the list scheduling algorithm is applied to the problem $Fm \mid k_{m-1} = 1, k_m = k \geq 2 \mid C_{\max}$, they get the bound $C_{\max} / C_{\max}^{optimum} \leq m + 1 - \frac{1}{k}$. The authors propose two algorithms for the $F2 \mid k_1 = k_2 = k \geq 2 \mid C_{\max}$ problem. First one is based on the list scheduling algorithm whereas the other one is based on the LPT rule.

Gupta and Tunc (1991) study the two-stage hybrid flowshop problem with multiple identical machines in second stage. They propose several lower bounds, polynomial bounded approximate algorithms and an improved branch-and-bound algorithm. The authors form a global lower bound by assigning the maximum of the proposed lower bounds. This global lower bound on makespan is used as surrogates for minimum makespan values for large problems, so the effectiveness of the proposed heuristic algorithms is calculated. They develop two polynomial bounded approximate algorithms, and then these algorithms are used to improve the efficiency of an existing branch-and-bound algorithm by assuming the makespan obtained from one of the two heuristic algorithms as initial upper bound. Their computational experiments show the satisfactory performance of their algorithms. Gupta et al. (1997) study the scheduling of two-stage hybrid flowshop problem in which there are many identical machines at the first stage. The authors present several lower bounds, a branch-and-bound algorithm and constructive heuristics. The proposed lower bounds and a dominance rule are used to restrict the size of the search tree. Besides, it is pointed out that using one of the heuristic algorithms in the literature as an initial upper bound helps shorten the search in the tree. Computational experiments show the efficiency of the algorithms.

Oğuz et al. (2003) propose heuristic algorithms for a two-stage hybrid flowshop scheduling problem. The heuristics are based on some priority rules. In the first part of the heuristics, they construct the processing order for the jobs by using various sequencing rules like SPT, LPT, etc. In the second part, the schedule is formed for the obtained sequence. The lower bounds are developed to evaluate the performance of the heuristics. Haouari and M'Hallah (1997) study the two-stage hybrid flowshop problem with several identical machines per stage. They suggest a two phase heuristic. Firstly a feasible solution is computed by using a simple heuristic that is easy to implement and then simulated annealing and tabu search are employed to improve the initial solution. The results are compared with a new derived lower bound. Their computational experiments show the efficiency of the proposed heuristics.

### 3.3.2 Multi-Stage Hybrid Flowshop

Pioneer work on the multi-stage hybrid flowshop scheduling problem is done by Wittrock (1985). He develops a heuristic periodic algorithm where small set of jobs is scheduled and the schedule is repeated many times. However, it is difficult to adapt periodic scheduling to practical scheduling environments. In another study of Wittrock (1988), the author presents a non-periodic algorithm. He suggests decomposing the scheduling problem into three sub-problems: machine allocation, sequencing and timing. The algorithm employs the LPT heuristic for machine allocation and the workload approximation heuristic for sequencing. For the first time, Brah and Hunsucker (1991) propose several lower bounds and a branch-and-bound algorithm for multi-stage hybrid flowshop scheduling problem to minimize makespan. They also develop elimination rules to utilize along with the derived lower bounds to increase the efficiency of the algorithm. Jin et al. (2006) study the same problem. They propose a number of lower bound and two metaheuristic algorithms that are based on the simulated annealing and shop partitioning. Their computational experiments provide satisfactory performance for two, three and five stage problems with maximum 10 machines at each stage.

## 3.4 Hybrid Flowshop Scheduling Problem with Dedicated Machines

Dedicated machines are specialized for the execution of certain type of jobs. Oğuz et al. (1997) study a problem in which first stage includes two dedicated machines and second stage contains a common machine. In their manufacturing environment, two different products are initially processed by independent dedicated machines and finally jobs flow through a common machine such as an inspection and testing station. The authors prove that the $F3 \mid T = 2 \mid C_{\max}$ problem is *NP*-hard. They study two polynomially solvable cases of the problem and present their solution procedure. They propose a heuristic algorithm that is based on the Johnson’s rule and analysis the worst-case bound for this heuristic. Computational experiments show the efficiency of the proposed algorithm. Riane et al. (1998) study hybrid three-stage flowshop problem with one machine in the first and third stages and two dedicated machines in stage two. They propose two heuristic algorithms which one of them is dynamic programming-based and the other one is branch-and-bound-based. Their computational experiments which are conducted for varying job numbers from 10 to 130 show the satisfactory performance of their algorithms. Riane et al. (2002) study hybrid three-stage flowshop problem with one machine in the first and two dedicated machines in the second stage. The authors point out that the problem is *NP*-hard and presents four polynomially solvable cases. They propose an exact algorithm based on a dynamic program and three heuristic algorithms for large scale problems. Two of the heuristics are based on the Johnson’s rule and the other heuristic is a greedy heuristic. Their computational experiments show the efficiency of the algorithms.

In this section, 60 journal articles are reviewed according to the machine environment and objective function, makespan. The most closely related study to ours is Oğuz et al. (1997). They study the same problem with two-stage flow lines, unlike ours, in flowshop machine environment.

In this study, we extend their work by considering the problem with three-stage flow lines in hybrid flowshop machine environment. There are many studies in scheduling literature that deal with makespan minimization problem in hybrid flowshop environment; however, scheduling in a three-stage dedicated hybrid flowshop with a common third-stage is studied first time with this work, to the best of our knowledge.

# CHAPTER 4

# BRANCH-AND-BOUND ALGORITHM FOR THE REVERSED PROBLEM

Our problem of scheduling in a three-stage dedicated hybrid flowshop with a common third stage is a *NP*-hard problem of minimizing makespan. In Chapter 2, we develop a mathematical model to find optimal solutions and in this chapter, we present a branch-and-bound algorithm to solve the problem exactly.

## 4.1 The Reversed Problem

In the reversed problem, the direction of the job flow is reversed so that the jobs flow from the common machine to their related production lines according to their types. As it is illustrated by Figure 4.1, in stage 1, the first operations, last operations in the original problem, of the jobs are processed on the common machine regardless of the job type. The next operations of type-1 jobs are performed on the first flow line that contains two serial machines in stage 2 and 3, respectively. The last operations, first operations in the original problem, of type-2 jobs are completed on one of the identical parallel machines.

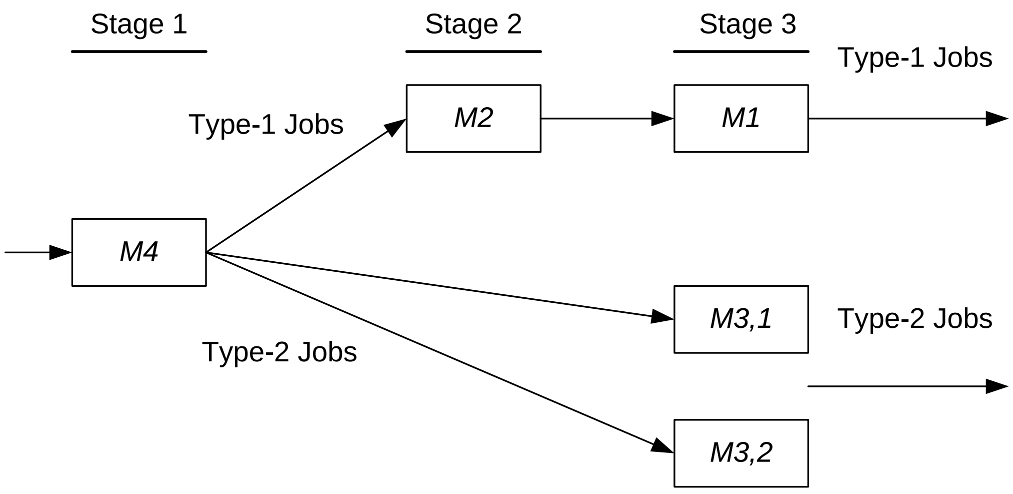

**Figure 4.1: The Reversed Problem**

**Proposition 1:** In the three-stage dedicated hybrid flowshop with a common third-stage, makespan minimization problem and its reverse are equivalent.

**Proof:** Consider any feasible schedule $S$ for the original problem. By right-shifting the jobs so that the makespan of the schedule $S$ will not increase will lead to a feasible schedule $S^{'}$ for the reverse problem. The feasibility of $S$ ensures that $S^{'}$ is feasible. Moreover, the makespans for these two schedules in their respective problems are identical. That is, $C_{\max}\left(S\right)=C_{\max}\left(S^{'}\right)$. Similarly, any feasible schedule for the reverse problem converts into a feasible schedule for the original problem with the same makespan. Thus, the two problems, original and reverse, are equivalent.

## 4.2 Branch-and-bound Algorithm for the Reversed Problem

In this section, we propose a branch-and-bound algorithm for the reversed problem to find exact solution to the problem addressed in this research. Since constructing branch-and-bound algorithm for reversed problem is easier than constructing for original problem, our branch-and-bound procedure is designed for the reversed problem to take the advantage of scheduling problems are symmetric with respect to time-reversal as shown in Proposition 1.

In order to use the branch-and-bound algorithm, the problem must be described as a tree in which each node represents an allocation of some of the jobs. The first node in the tree structure is called root node. It can be thought as starting node. From this node $n$ branches corresponding to $n$ possible jobs that can be assigned to the first position in the sequence are originated. From each of these nodes $n-1$ branches are produced which corresponding to $n-1$ possible jobs that can be assigned to the second position in the sequence, etc. At level $s$ of the tree, the decision about assignment of the $s^{th}$ job of the sequence is made and a complete schedule is reached at level $n$. The node having the lowest lower bound is chosen to branch at any level $s \leq n$. After a complete schedule is obtained at level $n$, we backtrack to level $n-1$ and continue from this stage. Thus, there are $n!$ feasible solutions and $1+n+n(n-1)+...+n!$ nodes in the tree unless any of nodes are fathomed. We fathom the node if the associated lower bound is greater than or equal to the upper bound. Upper bound is updated whenever a complete schedule with $n$ jobs and smaller lower bound is found. We terminate when there is no unfathomed or nonbranched node. And, the latest updated upper bound is the optimal solution.

*Depth-first search* method is utilized to guide in the branch-and-bound tree. According to this strategy, the branch of the tree goes down by dealing only one job at each level. We reach a complete schedule at the bottom of the tree and then we backtrack. We favor this strategy because of its relatively low memory requirement.

The following notation is needed for the description of the lower bounds.

$J_s$ = the ordered set of scheduled jobs at any arbitrary node, say $i$.

$J_u$ = the set of unscheduled jobs at any arbitrary node, say $i$.

$J_u^f$ = the set of unscheduled jobs that will be processed on flowshop environment at any arbitrary node, say $i$.

$J_u^p$ = the set of unscheduled jobs that will be processed on parallel machine environment at any arbitrary node, say $i$.

$LB_i$ = the lower bound value for the makespan at node $i$.

$P_{jk}$ = processing time of job $j$ at stage $k$.

$T_k(J_s)$ = the time at which machine at stage $k$ completes processing on the jobs in set $J_s$.

Then, a lower bound on the makespan at node $i$ is calculated as below

$$LB_i = \max\left\{LB_i^1,\ LB_i^2,\ LB_i^3,\ LB_i^4,\ LB_i^5\right\},$$

where

$$LB_i^1 = T_4(J_s) + \sum_{j \in J_u^f} P_{j,3} + \min_{j \in J_u^f}\left\{P_{j,2} + P_{j,1}\right\} \tag{4.1}$$

$$LB_i^2 = T_2(J_s) + \sum_{j \in J_u^f} P_{j,2} + \min_{j \in J_u^f} \left\{ P_{j,1} \right\} \tag{4.2}$$

$$LB_i^3 = T_1(J_s) + \sum_{j \in J_u^f} P_{j,1} \tag{4.3}$$

$$LB_i^4 = T_4(J_s) + \sum_{j \in J_u^p} P_{j,3} + \min_{i,j \in J_u^p} \left\{ P_{j,1} \right\} \tag{4.4}$$

$$LB_i^5 = \max\left\{ T_4(J_s), \min\left\{ T_{3,1}(J_s),\ T_{3,2}(J_s) \right\} \right\} + \max\left\{ \begin{array}{l} \max_{i,j \in J_u^p} \left\{ P_{j,1} \right\}, \\ \dfrac{\sum_{i,j \in J_u^p} P_{j,1} - \max\left\{ \begin{array}{l} 0, \max\left\{ T_4(J_s), T_{3,1}(J_s),\ T_{3,2}(J_s) \right\} \\ - \max\left\{ T_4(J_s), \min\left\{ T_{3,1}(J_s),\ T_{3,2}(J_s) \right\} \right\} \end{array} \right\}}{2} \end{array} \right\} \tag{4.5}$$

The stepwise description of our branch-and-bound algorithm is given below:

Step 1. Initialization

1.1. Let $k = 0,\ J_s = \varnothing,\ s(J_u) = n,\ LB_i = 0$ for all $i$ in the tree.

1.2. Compute upper bound, *UB*, from proposed heuristic algorithm.

Step 2. Partial Schedule Arrangement

2.1. Set $k = k + 1.$

2.2. If $k > n,$ then go to Step 6.

Step 3. Lower Bounding

3.1. Compute $LB_i$ for all unscheduled jobs.

3.2. If $LB_i > UB$, fathom node $i$.

3.2. If all nodes are pruned, then go to Step 6.

Step 4. Branching

4.1. At each level branch from the node having the lowest lower bound

4.2. If $k = n,$ then go to Step 5 else go to Step 2.

Step 5. Upper Bound Updating

If complete schedule has a better makespan value, then updates UB.

Step 6. Backtracking

6.1. Set $k = k - 1.$

6.2. If $k = 0,$ then go to Step 7 else go to Step 3.

Step 7. Termination

7.1. Terminate the procedure when there is no unfathomed or branched node.

7.2. The last upper bound is optimal solution to our problem

A numerical example that illustrates the implementation of branch-and-bound algorithm for reversed problem is given in APPENDIX A.

# CHAPTER 5

# PROPOSED HEURISTIC ALGORITHMS

Since the three-stage dedicated hybrid flowshop problem is *NP*-hard, an optimizing algorithm that runs in polynomial-time cannot exist. In this section, we propose heuristic algorithms; one of them is based on the six different sequencing rules and the other one is based on the branch-and-bound algorithm, to schedule jobs on the machines at all stages. It is important to evaluate the performance of these heuristic procedures, because we employ them as an upper bound in our exact solution approaches, i.e., branch-and-bound algorithm and mathematical model. Moreover, when the computational effort to obtain an exact solution with branch-and-bound algorithm or mathematical model is prohibitive, these heuristics provide good quality solution at little computational effort. In order to evaluate how good the proposed heuristics algorithms, we consider the optimum value that is obtained by exact solution approaches for small size problems and the global lower bound that is generated by our proposed four lower bounds.

## 5.1 Sequencing Rules and the Sequencing-rules Based Heuristic

### 5.1.1 Sequencing Rules

<u>Rule *A:*</u>

Step 1: a) Sequence the jobs of the two stage of the first flow line by applying Johnson Algorithm with processing times $p_{j,1}$ and $p_{j,2}$; and

b) Sequence the jobs of the parallel machining shop by applying Shortest Processing Time List Rule with processing times $p_{j,1}$.

Step 2: Sequence the jobs on the common stage in an ascending order of their completion times (earliest release time) at the previous stage.

In this rule, our motivation is to finish the first and second operations of all jobs as soon as possible.

The computational time complexity of the sequencing rule $A$ is $O(n_1 \log n_1 + n_2 \log n_2)$. In Step 1 of the sequencing rule, both type-1 and type-2 jobs have to be sorted. These operations take $O(n_1 \log n_1 + n_2 \log n_2)$ steps. Furthermore, Step2 can be done in $O(n_1 + n_2)$ time. Thus, we have $O(n_1 \log n_1 + n_2 \log n_2)$ complexity.

Rule *B:*

The Rule $B$ is similar to the previous one. Instead of SPT, LPT is employed to sequence the jobs of the parallel machining shop.

Step 1: a) Sequence the jobs of the two stage of the first flow line by applying Johnson Algorithm with processing times $p_{j,1}$ and $p_{j,2}$; and

b) Sequence the jobs of the parallel machining shop by applying LPT with processing times $p_{j,1}$.

Step 2: Sequence the jobs on the common stage in an ascending order of their completion times (earliest release time) at the previous stage.

The computational time complexity of the sequencing rule *B* is the same with the previous one, $O(n_1 \log n_1 + n_2 \log n_2)$**.** In Step 1 of the sequencing rule, both type-1 and type-2 jobs have to be sorted, which can be done in $O(n_1 \log n_1 + n_2 \log n_2)$ time. Moreover, operation in Step2 takes $O(n_1 + n_2)$ time. Thus, we have $O(n_1 \log n_1 + n_2 \log n_2)$ complexity.

*Rule C:*

The Rule *C* is based on the algorithm due to Campbell, Dudek and Smith (CDS), which is based on the repeated application of the Johnson algorithm. The underlying idea of this algorithm is to generate two fictitious two-stage flow shop problems for first flow line. These fictitious two-stage flow shop problems have the processing times $(p_{j,1}, p_{j,3})$ and $(p_{j,1} + p_{j,2}, p_{j,2} + p_{j,3})$, respectively. We then apply Johnson Algorithm to each of the two fictitious two-stage flow shop problems generated for first flow line, and finally select the better of two sequences. The Rule *C* has the following three steps:

Step 1: Sequence the jobs on the first two stages of first flow line by applying Johnson Algorithm with processing times $p_{j,1}$ and $p_{j,3}$. Sequence the jobs of the parallel machining shop by applying SPT with processing times $p_{j,1}$. Sequence the jobs on the common stage in an ascending order of their completion times (earliest release time) at the previous stage. Let $C_{\max}^{1}$ be the makespan of the resulting schedule.

Step 2: Sequence the jobs on the first two stages of first flow line by applying Johnson Algorithm with processing times $p_{j,1} + p_{j,2}$ and $p_{j,2} + p_{j,3}$.

Sequence the jobs of the parallel machining shop by applying SPT with processing times $p_{j,1}$. Sequence the jobs on the common stage in an ascending order of their completion times (earliest release time) at the previous stage. Let $C_{\max}^{2}$ be the makespan of the resulting schedule.

Step 3: If $C_{\max}^{1} \le C_{\max}^{2}$, select schedule obtained in Step 1 for implementation, otherwise, select schedule obtained in Step 2.

The computational time complexity of the sequencing rule *C* is $O\left(2\left(n_1 \log n_1 + n_2 \log n_2\right)\right)$. Since we execute the Step 1 and Step 2 of the rule *A* twice for each fictitious problem in rule *C*, its complexity is $O\left(2\left(n_1 \log n_1 + n_2 \log n_2\right)\right)$.

<u>Rule *D:*</u>

Step 1: Sequence the jobs on the first two stages of first flow line by applying Johnson Algorithm with processing times $p_{j,1}$ and $p_{j,1}$. Sequence the jobs of the parallel machining shop by applying LPT with processing times $p_{j,1}$. Sequence the jobs on the common stage in an ascending order of their completion times (earliest release time) at the previous stage. Let $C_{\max}^{1}$ be the makespan of the resulting schedule.

Step 2: Sequence the jobs on the first two stages of first flow line by applying Johnson Algorithm with processing times $p_{j,1} + p_{j,2}$ and $p_{j,2} + p_{j,3}$. Sequence the jobs of the parallel machining shop by applying LPT with processing times $p_{j,1}$. Sequence the jobs on the common stage in an ascending order of their completion times (earliest release time) at the previous stage. Let $C_{\max}^{2}$ be the makespan of the resulting schedule.

Step 3: If $C_{\max}^{1} \le C_{\max}^{2}$, select schedule obtained in Step 1 for implementation, otherwise, select schedule obtained in Step 2.

The computational time complexity of the sequencing rule *D* is the same with the previous one. The sorting operation in Step 1 takes $O\left(n_1 \log n_1 + n_2 \log n_2\right)$ time. Furthermore, Step2 can be done in $O\left(n_1 + n_2\right)$ time. Since we execute the Step 1 and Step 2 twice for each fictitious problem in rule *D*, its complexity is $O\left(2\left(n_1 \log n_1 + n_2 \log n_2\right)\right)$.

<u>Rule *E:*</u>

Step 1: a) Sequence the jobs on the first two stages of first flow line by applying Shortest Processing Times Rule with processing times $p_{j,1} + p_{j,2}$.

b) Sequence the jobs of the parallel machining shop by applying SPT with processing times $p_{j,1}$.

Step 2: Sequence the jobs on the common stage in an ascending order of their completion times (earliest release time) at the previous stage.

The sequencing rule *E* has the same computational time complexity with the sequencing rules *A* and *B*, $O\left(n_1 \log n_1 + n_2 \log n_2\right)$. In Step 1 of the sequencing rule, both type-1 and type-2 jobs have to be sorted. These operations take

$O(n_1 \log n_1 + n_2 \log n_2)$ steps. Furthermore, Step2 can be done in $O(n_1 + n_2)$ time. Thus, we have $O(n_1 \log n_1 + n_2 \log n_2)$ complexity.

<u>Rule *F:*</u>

Step 1: a) Sequence the jobs on the first two stages of first flow line by applying LPT with processing times $p_{j,1} + p_{j,2}$.

b) Sequence the jobs of the parallel machining shop by applying LPT with processing times $p_{j,1}$.

Step 2: Sequence the jobs on the common stage in an ascending order of their completion times (earliest release time) at the previous stage.

The computational time complexity of the sequencing rule *F* is computed as follows. The sorting operation in Step 1 takes $O(n_1 \log n_1 + n_2 \log n_2)$ time. Furthermore, the time complexity of Step2 is $O(n_1 + n_2)$. Thus, we have $O(n_1 \log n_1 + n_2 \log n_2)$ complexity.

### 5.1.2 The Sequencing-rules Based Heuristic

We develop a heuristic algorithm based on the six different sequencing rules to schedule jobs on the machines at all stages.

Step 1: Apply sequencing rules through Rule *A* to Rule *F*, respectively; and calculate their associated makespan value.

Step 2: Select the rule which gives the minimum makespan value to be implemented.

## 5.2 The Branch-and-bound Based Heuristic

Branch-and-bound is a well-known algorithm for finding an optimal solution to optimization problems. However, the computational effort to obtain an exact solution with branch-and-bound algorithm is prohibitive for large scale problem. For that reason, we propose a heuristic procedure that is based on the branch-and-bound algorithm.

In Section 4.2, the proposed branch-and-bound algorithm is explained in detail. In order to use this algorithm, the problem is described as a tree and partial solutions are arranged at each level of this tree. At each stage of the solution, an allocation of an unscheduled job to the sequence is done. A complete schedule is obtained when all jobs are scheduled. This is our initial solution. After the computation of the initial solution, the backtracking procedure starts.

The branch-and-bound based heuristic does not take into account the backtracking part of the branch-and-bound algorithm. The initial solution of the algorithm is accepted as a heuristic solution.

A numerical example that illustrates the implementation of the branch-and-bound based heuristic is given in APPENDIX B.

## 5.3 Lower Bounds

In this section, we derive four lower bounds for the makespan of an optimal schedule, which will be used in evaluating the performance of the proposed heuristic algorithms.

### 5.3.1 Lower Bound 1

First lower bound is obtained by assuming that the job with the minimum processing time for the third operation determines the makespan. The expression in (5.1) states that the processing of the operations, before the last operation on the common machine, for all jobs cannot be completed before total processing time of type-1 jobs at second stage plus minimum processing time of type-1 jobs at first stage or total processing time of type-1 jobs at first stage plus minimum processing time of type-1 jobs at second stage or half of total processing time of type-2 jobs at first stage;

$$\max\left\{\sum_{j\in J_1} p_{j,2}+\min_{j\in J_1}\left\{p_{j,1}\right\},\ \sum_{j\in J_1} p_{j,1}+\min_{j\in J_1}\left\{p_{j,2}\right\},\ \sum_{j\in J_2}\frac{p_{j,1}}{2}\right\}, \tag{5.1}$$

where $p_{j,i}$ is the processing time of $j$ at stage $i$. Therefore, the makespan of any schedule cannot be less than the expression in (5.1) plus the minimum processing time for the third operation. Thus, we obtain the first lower bound as

$$LB_1=\max\left\{\sum_{j\in J_1} p_{j,2}+\min_{j\in J_1}\left\{p_{j,1}\right\},\ \sum_{j\in J_1} p_{j,1}+\min_{j\in J_1}\left\{p_{j,2}\right\},\ \sum_{j\in J_2}\frac{p_{j,1}}{2}\right\} +\min_{j\in J}\left\{p_{j,3}\right\} \tag{5.2}$$

### 5.3.2 Lower Bound 2

We derive the second lower bound by assuming that the third stage operates continuously without any idle time between jobs, and the job which satisfies the condition,

$$\min\left\{\min_{j\in J_1}\left\{p_{j,1}+p_{j,2}\right\},\min_{j\in J_2}\left\{p_{j,1}\right\}\right\} \tag{5.3}$$

determines the makespan. The expression in (5.3) gives us the minimum processing time of the operations before the last operation is processed on the common machine. Thus, the second lower bound can be derived as

$$LB_2 = \min\left\{\min_{j\in J_1}\left\{p_{j,1}+p_{j,2}\right\},\min_{j\in J_2}\left\{p_{j,1}\right\}\right\}+\sum_{j\in J} p_{j,3}. \tag{5.4}$$

### 5.3.3 Lower Bound 3

Consider only type-1 jobs. For any sequence, we assume that the job with minimum processing time for the third operation determines the makespan.
Thus, we can establish our third lower bound as

$$LB_3 = \max_{j\in J_1}\left\{\sum_{i=1}^{j} p_{[i],1}+\sum_{i=j}^{n_1} p_{[i],2}\right\}+\min_{j\in J}\left\{p_{j,3}\right\}, \tag{5.5}$$

where $[i]$ is the job in the $i^{th}$ position of the sequence and $n_1$ is the number of type-1 jobs.

$LB_3$ can be rewritten by changing the bounds of summation for first term of (5.5) as

$$LB_3 = \max_{j\in J_1}\left\{\sum_{i=1}^{j} p_{[i],1}+\sum_{i=1}^{n_1} p_{[i],2}-\sum_{i=1}^{j-1} p_{[i],2}\right\}+\min_{j\in J}\left\{p_{j,3}\right\} \tag{5.6}$$

$$= \max_{j \in J_1} \left\{ \sum_{i=1}^{j} p_{[i],1} - \sum_{i=1}^{j-1} p_{[i],2} \right\} + \sum_{i=1}^{n_1} p_{[i],2} + \min_{j \in J} \left\{ p_{j,3} \right\}. \tag{5.7}$$

Note that the second term in the equation above is constant and independent from the sequence, and the first term,

$$\max_{j \in J_1} \left\{ \sum_{i=1}^{j} p_{[i],1} - \sum_{i=1}^{j-1} p_{[i],2} \right\} \tag{5.8}$$

gives the total idle time at the second stage. The expression in (5.8) has the form, which is same one as the Johnson's with processing times at the first and second stages, respectively, and the idle time can be minimized by sequencing job $k$ preceding job $l$ if $\min\left\{p_{k,1}, p_{l,2}\right\} \leq \min\left\{p_{l,1}, p_{k,2}\right\}$.

Therefore,

$$LB_3 = C^*_{\max}\left( \underset{j \in J_1}{JA} \left( p_{j,1}, p_{j,2} \right) \right) + \min_{j \in J} \left\{ p_{j,3} \right\}, \tag{5.9}$$

where $C^*_{\max}\left( \underset{j \in J_1}{JA} \left( p_{j,1}, p_{j,2} \right) \right)$ is the makespan of the two-stage problem, which is obtained by Johnson's Algorithm with processing times $p_{j,1}$ and $p_{j,2}$.

### 5.3.4 Lower Bound 4

Another lower bound is obtained by reducing the problem to classical two machine flowshop problem by dividing the job processing time at first stage by the number of machines, i.e., two identical parallel machines. So, the new processing time of job $j$ at stage 1 is defined as $\frac{p_{j,1}}{2}$. Then the problem with two-machine, one is at the first stage and the other one is at the third stage, is solved by Johnson's algorithm (Johnson (1954)). Thus, the fourth lower bound can be derived as

$$LB_4 = C^*_{\max}\left( \underset{j \in J_2}{JA} \left( \frac{p_{j,1}}{2}, p_{j,3} \right) \right) \tag{5.10}$$

From the above four lower bounds, it follows that the makespan of an optimal schedule $C^*_{\max}$ cannot be less than the global lower bound $GLB$. That is,

$$C^*_{\max} \geq GLB = \max_{i=1,\ldots,4} \{LB_i\} \tag{5.11}$$

## 5.4 A Numerical Example to Illustrate the Lower Bounds and Sequencing-rules Based Heuristic

Suppose that we have six jobs with the processing times given in Table 5.1. The first three jobs are type-1 jobs; others are type-2 jobs.

**Table 5.1: Processing Times for the Numerical Example**

| | **Processing Times** | | |
|---|---|---|---|
| **Jobs** | **Stage 1** | **Stage 2** | **Stage 3** |
| **1** | 7 | 1 | 2 |
| **2** | 3 | 3 | 6 |
| **3** | 2 | 6 | 5 |
| **4** | 1 | 0 | 8 |
| **5** | 3 | 0 | 4 |
| **6** | 7 | 0 | 4 |

### 5.4.1 Lower Bounds

The lower bounds and global lower bound are computed as follows:

<u>*Lower Bound* 1</u>:

$$LB_1 = \max\left\{\sum_{j\in J_1} p_{j,2} + \min_{j\in J_1}\left\{p_{j,1}\right\}, \sum_{j\in J_1} p_{j,1} + \min_{j\in J_1}\left\{p_{j,2}\right\}, \sum_{j\in J_2} \frac{p_{j,1}}{2}\right\}$$

$$+\min_{j\in J}\left\{p_{j,3}\right\}$$

$$\Rightarrow$$

$$LB_1 = \max\left\{(1+3+6)+2,\ (7+3+2)+1,\ \frac{1+3+7}{2}\right\}+2=15.$$

<u>*Lower Bound* 2</u>:

$$LB_2 = \min\left\{\min_{j\in J_1}\left\{p_{j,1}+p_{j,2}\right\}, \min_{j\in J_2}\left\{p_{j,1}\right\}\right\} + \sum_{j\in J} p_{j,3}$$

$$\Rightarrow$$

$$LB_2 = \min\left\{(3+3),1\right\} + 29 = 30.$$

<u>*Lower Bound* 3</u>:

$$LB_3 = C^*_{\max}\left(\underset{j\in J_1}{JA}\left(p_{j,1}, p_{j,2}\right)\right) + \min_{j\in J_1}\left\{p_{j,3}\right\}$$

$$\Rightarrow$$

$$LB_3 = 13+2=15, \text{ where } \underset{i\in J_1}{JA} : 3-2-1.$$

*Lower Bound* 4:

$$LB_4 = C^*_{\max}\left(\underset{j\in J_2}{JA}\left(\frac{p_{j,1}}{2}, p_{j,3}\right)\right)$$

$$\Rightarrow$$

$$LB_4 = 17, \text{ where } \underset{i\in J_2}{JA} : 4-5-6.$$

Therefore, the global lower bound $GLB$

$$LB = \max\{15,\ 30,\ 15,\ 17\} = 30.$$

**5.4.2 Sequencing Rules**

*Rule A*:

By sequencing the jobs of the two stage of the first flow line by applying Johnson Algorithm with processing times $p_{j,1}$ and $p_{j,2}$, we get $\underset{i\in J_1}{JA} : 3-2-1$, and by sequencing the jobs of the parallel machining shop by applying SPT with processing times $p_{j,1}$, we get $\underset{i\in J_2}{SPT} : 4-5-6$. The Gantt chart for the schedule obtained by sequencing rule $A$ is given in Figure 5.1.

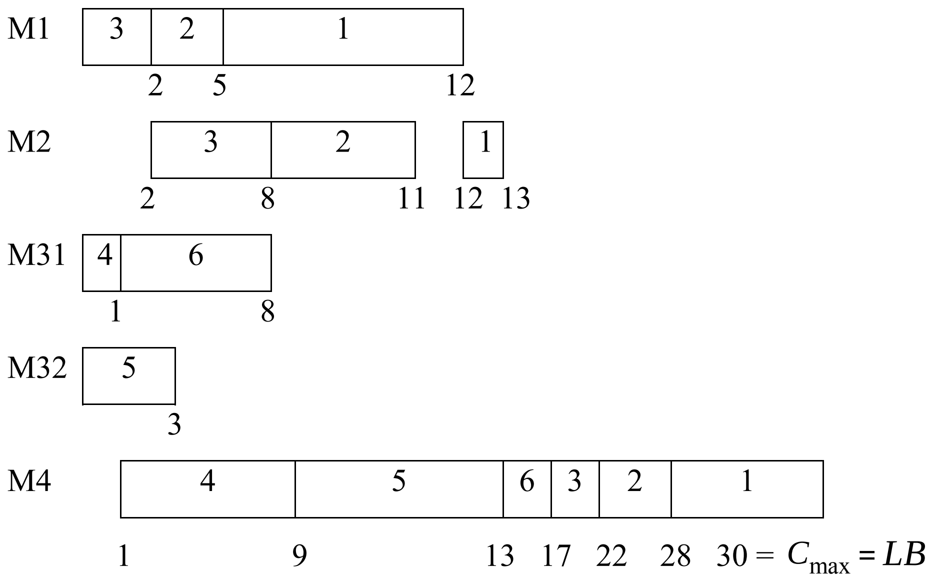

**Figure 5.1: Schedule Obtained by Sequencing Rule *A***

<u>*Rule B*</u>:

The sequencing rule *B* is similar to the sequencing rule *A*. The sequence of the jobs of the two stage of the first flow line is the same with the previous one that is determined for rule *A*, $\underset{i\in J_1}{JA}: 3-2-1$, and by sequencing the jobs of the parallel machining shop by applying LPT with processing times $p_{j,1}$, we get $\underset{i\in J_2}{LPT}: 6-5-4$. The related Gantt chart is given in Figure 5.2.

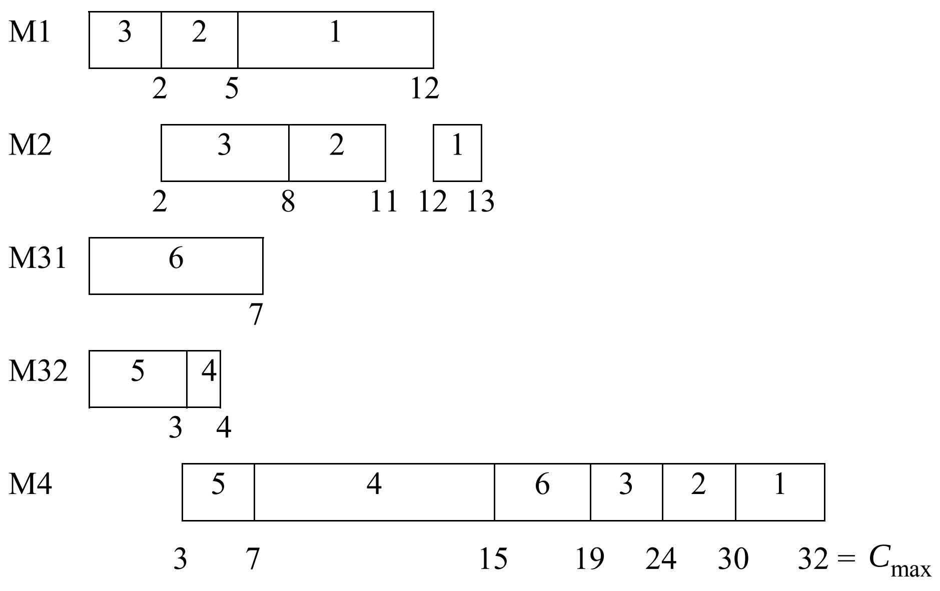

**Figure 5.2: Schedule Obtained by Sequencing Rule *B***

*Rule C*:

By sequencing the jobs of the two stage of the first flow line by applying Johnson Algorithm with processing times $p_{j,1}$ and $p_{j,3}$, we get $\underset{i \in J_1}{JA}: 3-2-1$, and by sequencing the jobs of the parallel machining shop by applying SPT with processing times $p_{j,1}$, we get $\underset{i \in J_2}{SPT}: 4-5-6$. The $C^1_{\max}$ value according to the given sequences is 30.

In the second part of the rule, the sequence of the jobs on the first two stages of first flow line is determined by applying Johnson Algorithm with processing times $p_{j,1}+p_{j,2}$ and $p_{j,2}+p_{j,3}$, $\underset{i \in J_1}{JA}: 2-3-1$, and the sequence of the jobs of the parallel machining shop is the same with the one which is determined in the first part of the heuristic, $\underset{i \in J_2}{SPT} : 4-5-6$. And, finally by sequencing the jobs on the third stage in an ascending order of their completion times at the previous stage, the $C^2_{\max}$ value is computed as 32.

Since $C^1_{\max} < C^2_{\max}$, we select the schedule obtained in first part of the heuristic for implementation. The schedule obtained by sequencing rule *C* is illustrated by the Gantt chart in Figure 5.3.

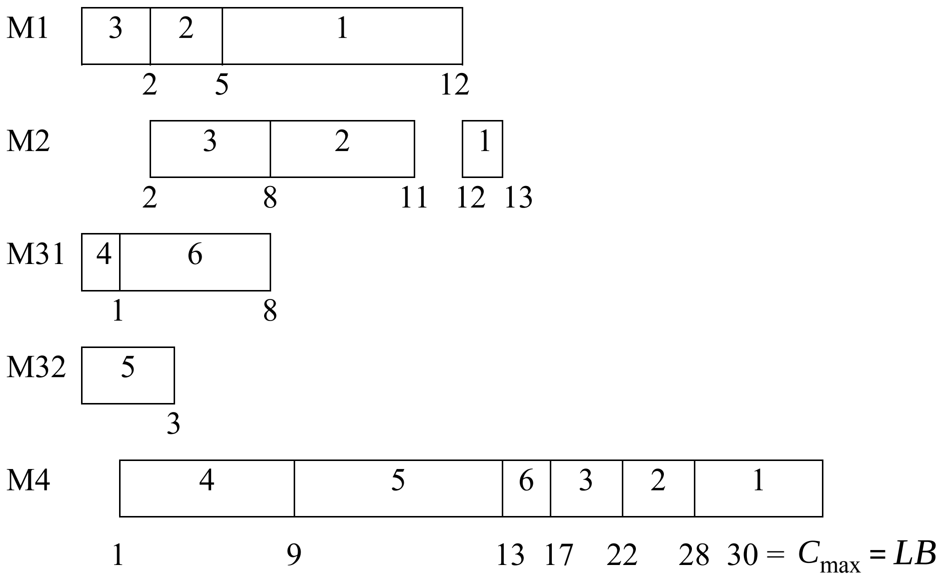

**Figure 5.3: Schedule Obtained by Sequencing Rule *C***

<u>*Rule D*</u>:

The sequence of the jobs for the first two stages of the first flow line is the same for the first and second part of the rule with the previous one that is determined for rule *C*, $\underset{i \in J_1}{JA}: 3-2-1$ and $\underset{i \in J_1}{JA}: 2-3-1$, respectively. The sequence of the jobs of the parallel machining shop is determined by applying LPT with processing times $p_{j,1}$, $\underset{i \in J_2}{LPT}: 6-5-4$. Since $C_{\max}^1 = 32 = C_{\max}^2 = 32$, we can select either of the schedules obtained in the first part of the heuristic or in the second part for implementation. The Gantt chart given in Figure 5.4 illustrates the schedule obtained by sequencing rule *D*.

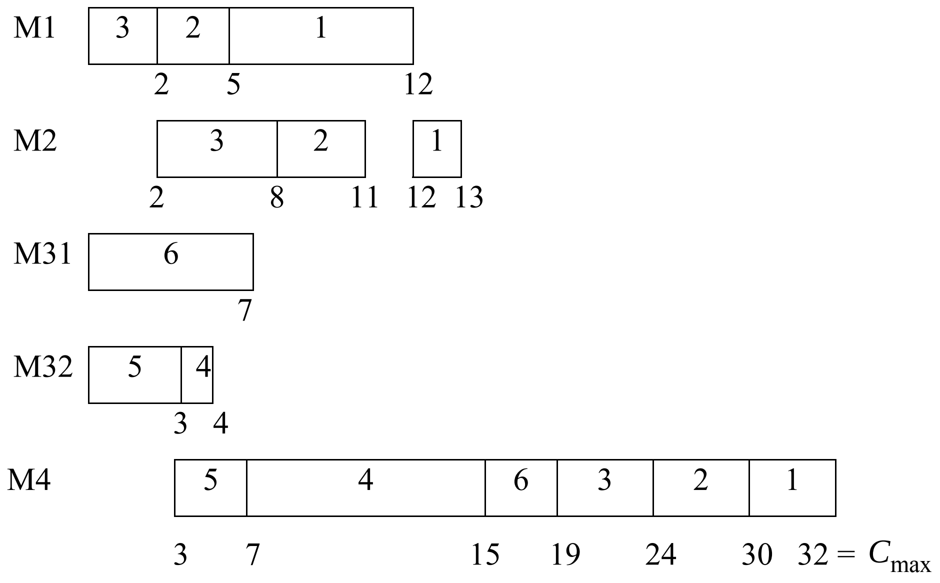

**Figure 5.4: Schedule Obtained by Sequencing Rule *D***

<u>*Rule E*:</u>

The sequence of the jobs of the on the first two stages of first flow line is determined by applying Shortest Processing Times Rule with processing times $p_{j,1} + p_{j,2}$, $\underset{i \in J_1}{SPT}: 2-3-1$, and by sequencing the jobs of the parallel machining shop by applying Shortest Processing Time List Rule with processing times $p_{j,1}$, we get $\underset{i\in J_2}{SPT}: 4-5-6$. The schedule obtained by sequencing rule *E* is illustrated by the Gantt chart in Figure 5.5.

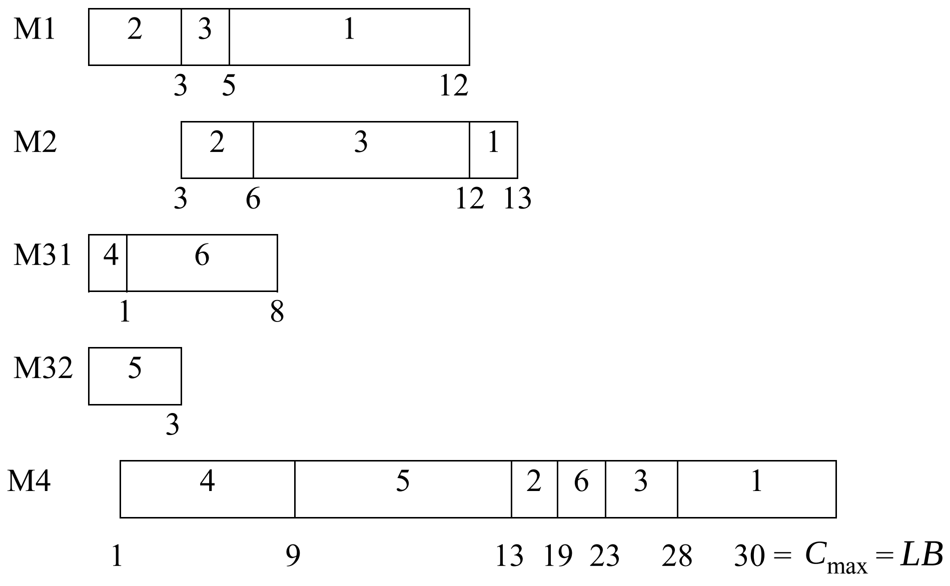

**Figure 5.5: Schedule Obtained by Sequencing Rule *E***

*<u>Rule F</u>:*

The sequence of the jobs of the on the first two stages of first flow line is determined by applying LPT with processing times $p_{j,1} + p_{j,2}$, $\underset{i\in J_1}{LPT}: 3-1-2$, and by sequencing the jobs of the parallel machining shop by applying LPT with processing times $p_{j,1}$, we get $\underset{i\in J_2}{LPT}: 6-5-4$. The Gantt chart for the schedule obtained by sequencing rule *F* is given in Figure 5.6.

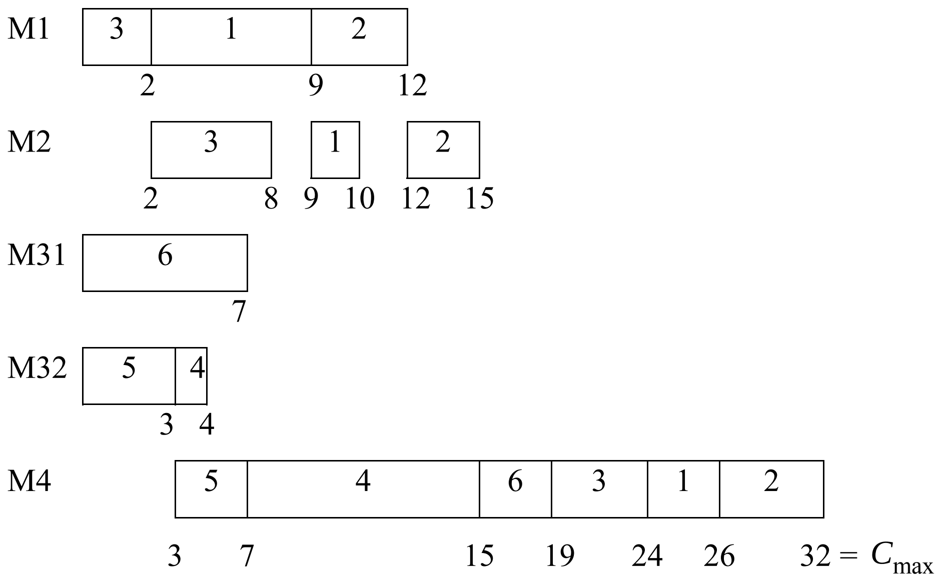

**Figure 5.6: Schedule Obtained by Sequencing Rule *F***

In this problem instance, makespan values for the schedules obtained by sequencing rules *A*, *C*, and *E* are equal to the global lower bound value, 30. Thus, we can conclude that the optimal makespan value is 30. Sequencing rules *B*, *D*, and *F* cannot yield optimal solution for the numerical example. Common part of the sequencing rules *A*, *C*, and *E* is the use of the SPT rule in order to sequence the jobs in both of the flow lines, unlike to the sequencing rules *B*, *D*, and *F*, which are based on the LPT rule. Someone can think that SPT rule is better than LPT rule for our problem. However, we cannot generalize this situation; it is true only for the problem instances which flowshop environment is dominant to the parallel machine environment in point of processing times, such that it is well known in the literature that SPT works better than LPT for flowshop environment. The sequencing rules *B*, *D*, and *F*, which are based on the LPT rule, can give better solutions than the sequencing rules *A*, *C*, and *E* with different data sets.

# CHAPTER 6

# COMPUTATIONAL EXPERIMENTS AND RESULTS

In this chapter, we discuss the results of our computational experiments to investigate the performance of our lower bounds on the makespan, proposed heuristic algorithms and branch-and-bound algorithm. Also, the effect of the certain parameters on the performance of our solution approaches is presented. We first explain our data generation scheme; then, the performance measures are presented. Finally, we discuss the results of our computational experiments.

## 6.1 Design of Experiments

The data generation procedures are given in this section. The following parameters are employed in the production of our data:

*Processing Time:* The integer processing times are generated from a discrete uniform distribution over [1, 20] in general. However, in order to evaluate the performance of our proposed solution approaches when one of the machine or production line is dominant to other one in terms of production time, we generate the production time from a discrete uniform distribution over various ranges. The possible cases and required conditions are given in Table 6.1.

Furthermore, these dominance cases are effective in generation of lower bounds for the original problem. Such that, the lower bound 1 is generated by taking the case 3 into account, the lower bound 2 is generated by taking the case 2 into account, the lower bound 3 is generated by taking the case 4 into account, the lower bound 4 is generated by taking the case 5 into account,

*Problem Size:* In order to see the effect of the number of type-1 and type-2 jobs, we use different number of jobs in our experiments. First of all, we define number of type-1 jobs, which are 10, 20, 30 and 50. Then, numbers of type-2 jobs are determined depending on these values; we generate number of type-2 jobs according to the following ratio:

$$\frac{\text{Number of Type-2 jobs}}{\text{Number of Type-1 jobs}} = \{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\} \qquad (6.1)$$

Size of the problem instances in our computational study and the related number of type-1 and type-2 jobs are given in Table 6.2.

**Table 6.1: Possible Dominance Cases Between Manufacturing Lines/Machines and Needed Conditions for These Cases**

| Case | Description | Condition |
|---|---|---|
| 1 | No operation is dominant (or dominated) | Randomly chosen data |
| 2 | Common machine dominates the other machines | $p(M_4)^* > \max\{p(M_1), p(M_2), p(M_{3,1}), p(M_{3,2})\}$ |
| 3 | Common machine is dominated by the other machines | $p(M_4) < \min\{p(M_1), p(M_2), p(M_{3,1}), p(M_{3,2})\}$ |
| 4 | Flowshop type jobs dominate parallel machine type jobs | $\sum p(M_1) > \frac{\sum\left(p(M_{3,1}) + p(M_{3,2})\right)}{2}$ |
| 5 | Parallel machine type jobs dominate flowshop type jobs | $\sum\left(p(M_{3,1}) + p(M_{3,2})\right) > 2 * \max\{p(M_1), p(M_2)\}$ |

* $p(M_i)$ indicates the processing time of a job on Machine *i* ( $M_i$ ) (See Figure 2.1).

For each combination of processing times, 10 problem instances are generated. The mathematical model is coded in GAMS 22.6 and the problem instances with up to 28 jobs are solved by using CPLEX solver under the time limit of one hour.

**Table 6.2: Size of the Problem Instances in the Computational Study**

| Number of Type-1 Jobs (n1) | Number of Type-2 Jobs (n2) | Number of Type-1 Jobs (n1) | Number of Type-2 Jobs (n2) |
|---|---|---|---|
| **10** | 2 | **30** | 6 |
| | 4 | | 12 |
| | 6 | | 18 |
| | 8 | | 24 |
| | 10 | | 30 |
| | 12 | | 36 |
| | 14 | | 42 |
| | 16 | | 48 |
| | 18 | | 54 |
| | 20 | | 60 |
| **20** | 4 | **50** | 10 |
| | 8 | | 20 |
| | 12 | | 30 |
| | 16 | | 40 |
| | 20 | | 50 |
| | 24 | | 60 |
| | 28 | | 70 |
| | 32 | | 80 |
| | 36 | | 90 |
| | 40 | | 100 |

The proposed heuristic algorithms, lower bounds and branch-and-bound algorithm are coded in Visual C++ 6.0, and all computational experiments are conducted on Intel Pentium IV 2400 MHz CPU with 256 MB memory PC under Windows XP operating system.

## 6.2 Performance Measures

The following performance measures are used to evaluate the performance of our solution methods and bounding procedures.

- The performance of global lower bound, *GLB*, is evaluated by using average and maximum value of its percentage deviation from optimal value, *OPT*, which is obtained by our exact solution methods (i.e. mathematical model, and branch-and-bound algorithm) is used. Thus, our performance measure is $\frac{OPT - GLB}{OPT} \times 100$

- The performance of heuristic algorithms is evaluated by using average and maximum value of its percentage deviation from optimal value which is obtained by our exact solution methods (i.e. mathematical model, and branch-and-bound algorithm). For the large size problems, which our exact solution approaches do not work, instead of optimal value we use global lower bound, *GLB*. Hence, our performance measure is

  $$\frac{H_i - OPT}{OPT} \times 100 \qquad \text{or} \qquad \frac{H_i - GLB}{GLB} \times 100$$

  where $H_i$ is the makespan value obtained by the scheduling rule *i*.

- Number of times, optimal solution is found in problem instances is another performance measure for the heuristic algorithms.

- Computation time is used to evaluate the performance of the branch-and-bound algorithm. Average and maximum values of computation times in CPU seconds are reported.

- Average and maximum numbers of nodes generated in the branch-and-bound tree are reported for each instance as another performance measure for our branch-and-bound algorithm.

- Number of unsolved problem instances in one hour is also used to evaluate the performance of the branch-and-bound algorithm.

## 6.3 Discussion of the Results

### 6.3.1 Global Lower Bound Performance

The average and maximum percent deviations of global lower bound from the optimal solution are reported in Table 6.3. As can be observed from the table, the problem instances up to the 28 jobs are considered in the experimentation. This is because the optimal solution is needed to investigate the global lower bound performance, and our exact solution methods are capable to solve problem up to with 28 jobs in a given time limit, which is one hour. Our results show that in general global lower bound performs better for small *n1*/*n2* ratio. For example, when *n1*=10 and dominance case 4, average deviations from optimal solution are 0.91% and 1.29% for *n1*/*n2*=0.2 and *n1*/*n2*=0.4, respectively. As Table 6.3 indicates, the global lower bound performs well and gives close solutions for all combinations of number of type-1, and type-2 jobs and also for every dominance cases. For example, when *n1*=10 and *n1*/*n2*=0.4, average deviations from optimal solution are 0.94%, 1.29%, and 2.03% for case 3, case 4 and case 5, respectively.

**Table 6.3: Global Lower Bound Performance- Percent Deviation from Optimum**

| n1 | n2/n1 | Case 1 | | n1 | n2/n1 | Case 2 | |
|---|---|---|---|---|---|---|---|
| | | $\left(OPT-GLB\right)/OPT\times 100$ | | | | $\left(OPT-GLB\right)/OPT\times 100$ | |
| | | Avg | Max | | | Avg | Max |
| | **0.2** | 1.2 | 3.5 | | **0.2** | 0 | 0 |
| | **0.4** | 1.52(2)* | 5.92 | | **0.4** | 0(2)* | 0 |
| | **0.6** | 0.12(5)* | 0.61 | | **0.6** | 0(4)* | 0 |
| | **0.8** | 0(6)* | 0 | | **0.8** | 0(5)* | 0 |
| **10** | **1.0** | 0(6)* | 0 | **10** | **1.0** | 0(5)* | 0 |
| | **1.2** | 0(7)* | 0 | | **1.2** | 0(8)* | 0 |
| | **1.4** | 0(7)* | 0 | | **1.4** | 0(7)* | 0 |
| | **1.6** | 0(8)* | 0 | | **1.6** | 0(8)* | 0 |
| | **1.8** | 0(9)* | 0 | | **1.8** | 0(7)* | 0 |
| **20** | **0.2** | 0(9)* | 0 | **20** | **0.2** | 0(8)* | 0 |
| | **0.4** | 0(8)* | 0 | | **0.4** | 0(6)* | 0 |
| **n1** | **n2/n1** | **Case 3** | | **n1** | **n2/n1** | **Case 4** | |
| | | $\left(OPT-GLB\right)/OPT\times 100$ | | | | $\left(OPT-GLB\right)/OPT\times 100$ | |
| | | **Avg** | **Max** | | | **Avg** | **Max** |
| | **0.2** | 0.71 | 2.29 | | **0.2** | 0.91 | 3.06 |
| | **0.4** | 0.94 | 2.62 | | **0.4** | 1.29 | 3.07 |
| | **0.6** | 2.03(5)* | 4.18 | | **0.6** | 1.22 | 3.46 |
| | **0.8** | 1.42(3)* | 2.54 | | **0.8** | 1.72(2)* | 3.47 |
| **10** | **1.0** | 1.14(5)* | 3.89 | **10** | **1.0** | 1.03(2)* | 1.84 |
| | **1.2** | 0.43(8)* | 0.86 | | **1.2** | 0.9(1)* | 2.18 |
| | **1.4** | (10)* | | | **1.4** | 1.16(1)* | 2.42 |
| | **1.6** | (10)* | | | **1.6** | 0.76(3)* | 2.47 |
| | **1.8** | (10)* | | | **1.8** | 0.64 | 1.67 |
| **20** | **0.2** | 0(8)* | 0 | **20** | **0.2** | 0.8(3)* | 1.84 |
| | **0.4** | 1.2(8)* | 1.69 | | **0.4** | 0.92(5)* | 1.46 |
| **n1** | **n2/n1** | **Case 5** | | | | | |
| | | $\left(OPT-GLB\right)/OPT\times 100$ | | | | | |
| | | **Avg** | **Max** | | | | |
| | **0.2** | 3.83 | 10.56 | | | | |
| | **0.4** | 2.03 | 6.71 | | | | |
| | **0.6** | 2.81(8)* | 4.17 | | | | |
| | **0.8** | (10)* | | | | | |
| **10** | **1.0** | (10)* | | | | | |
| | **1.2** | (10)* | | | | | |
| | **1.4** | (10)* | | | | | |
| | **1.6** | (10)* | | | | | |
| | **1.8** | (10)* | | | | | |
| **20** | **0.2** | (10)* | | | | | |
| | **0.4** | (10)* | | | | | |

(*) The numbers in parentheses denote the number of unsolved instances out of ten within one hour time limit.

## 6.3.2 Heuristic Algorithms Performances

Average and maximum percent deviations from the optimal and global lower bound are reported in Table 6.4 and Table 6.5, respectively. Furthermore, numbers of times, our heuristics find optimal solution is reported in Table 6.6.

**Table 6.4: Heuristic Algorithms Performance – Percent Deviation from Optimum**

| n1 | n2/n1 | Case 1 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | |
| | | Average | | | | | | | Maximum | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 7.47 | 7.47 | 0.86 | 0.86 | 7.90 | 7.90 | 5.45 | 21.26 | 21.26 | 7.87 | 7.87 | 20.28 | 20.28 | 14.39 |
| | 0.4 | 5.14(2)* | 7.43 | 0.69 | 3.70 | 4.30 | 7.03 | 12.17 | 14.48 | 14.48 | 5.52 | 6.67 | 19.08 | 19.08 | 31.03 |
| | 0.6 | 0.00(5)* | 4.46 | 0.00 | 2.63 | 0.00 | 2.63 | 2.90 | 0.00 | 6.70 | 0.00 | 4.04 | 0.00 | 4.04 | 7.58 |
| | 0.8 | 0.00(6)* | 5.54 | 0.00 | 3.95 | 0.00 | 3.95 | 2.44 | 0.00 | 7.30 | 0.00 | 5.83 | 0.00 | 5.83 | 6.15 |
| | 1.0 | 0.00(6)* | 6.57 | 0.00 | 3.41 | 0.00 | 3.52 | 4.38 | 0.00 | 8.37 | 0.00 | 4.89 | 0.00 | 4.89 | 6.67 |
| | 1.2 | 0.00(7)* | 3.30 | 0.00 | 2.17 | 0.00 | 2.17 | 4.98 | 0.00 | 5.45 | 0.00 | 3.18 | 0.00 | 3.18 | 8.33 |
| | 1.4 | 0.00(7)* | 4.98 | 0.00 | 4.61 | 0.00 | 4.23 | 5.56 | 0.00 | 5.68 | 0.00 | 5.68 | 0.00 | 4.84 | 14.71 |
| | 1.6 | 0.00(8)* | 5.78 | 0.00 | 2.29 | 0.00 | 2.29 | 1.88 | 0.00 | 7.11 | 0.00 | 3.56 | 0.00 | 3.56 | 3.75 |
| | 1.8 | 0.00(9)* | 4.67 | 0.00 | 0.39 | 0.00 | 0.39 | 42.41 | 0.00 | 4.67 | 0.00 | 0.39 | 0.00 | 0.39 | 42.41 |
| 20 | 0.2 | 8.84(9)* | 8.84 | 0.00 | 1.61 | 0.00 | 1.61 | 0.00 | 8.84 | 8.84 | 0.00 | 1.61 | 0.00 | 1.61 | 0.00 |
| | 0.4 | 0.34(8)* | 2.46 | 0.00 | 1.45 | 0.17 | 1.45 | 0.00 | 0.68 | 3.08 | 0.00 | 1.53 | 0.34 | 1.53 | 0.00 |
| n1 | n2/n1 | Case 2 | | | | | | | | | | | | | |
| | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | |
| | | Average | | | | | | | Maximum | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 0.03 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 1.11 | 0.28 | 0.28 | 0.00 | 0.00 | 0.00 | 0.00 | 2.81 |
| | 0.4 | 0.00(2)* | 1.44 | 0.00 | 0.80 | 0.00 | 0.80 | 1.59 | 0.00 | 3.33 | 0.00 | 1.55 | 0.00 | 1.55 | 3.43 |
| | 0.6 | 0.12(4)* | 1.07 | 0.00 | 0.45 | 0.00 | 0.45 | 1.58 | 0.73 | 2.22 | 0.00 | 0.89 | 0.00 | 0.89 | 3.53 |
| | 0.8 | 0.00(5)* | 1.11 | 0.00 | 0.71 | 0.00 | 0.71 | 0.68 | 0.00 | 2.59 | 0.00 | 1.23 | 0.00 | 1.23 | 1.64 |
| | 1.0 | 0.00(5)* | 1.53 | 0.00 | 0.80 | 0.00 | 0.80 | 1.02 | 0.00 | 2.25 | 0.00 | 2.25 | 0.00 | 2.25 | 2.09 |
| | 1.2 | 0.00(8)* | 1.13 | 0.00 | 1.13 | 0.00 | 1.13 | 1.48 | 0.00 | 1.70 | 0.00 | 1.70 | 0.00 | 1.70 | 1.83 |
| | 1.4 | 0.00(7)* | 1.22 | 0.00 | 0.99 | 0.00 | 0.99 | 0.88 | 0.00 | 1.66 | 0.00 | 1.41 | 0.00 | 1.41 | 1.28 |
| | 1.6 | 0.00(8)* | 1.16 | 0.00 | 1.16 | 0.00 | 1.16 | 1.36 | 0.00 | 1.97 | 0.00 | 1.97 | 0.00 | 1.97 | 1.53 |
| | 1.8 | 0.00(7)* | 1.27 | 0.00 | 0.55 | 0.00 | 0.55 | 0.79 | 0.00 | 2.28 | 0.00 | 0.81 | 0.00 | 0.81 | 1.22 |
| 20 | 0.2 | 0.00(8)* | 0.20 | 0.00 | 0.20 | 0.00 | 0.20 | 0.07 | 0.00 | 0.27 | 0.00 | 0.27 | 0.00 | 0.27 | 0.13 |
| | 0.4 | 0(6)* | 0.70 | 0.00 | 0.64 | 0.00 | 0.64 | 0.77 | 0.00 | 1.41 | 0.00 | 1.17 | 0.00 | 1.17 | 1.56 |

(*) The numbers in parentheses denote the number of unsolved instances out of ten within one hour time limit

**Table 6.4 (cont.)**

| n1 | n2/n1 | Case 3 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | |
| | | Average | | | | | | | Maximum | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 1.35 | 1.35 | 1.21 | 1.21 | 6.04 | 6.04 | 3.67 | 4.64 | 4.64 | 6.09 | 6.09 | 8.78 | 8.78 | 10.03 |
| | 0.4 | 2.18 | 2.18 | 1.69 | 1.69 | 4.18 | 4.18 | 5.20 | 5.38 | 5.38 | 7.59 | 7.59 | 7.80 | 7.80 | 15.86 |
| | 0.6 | 1.03(5)* | 1.03 | 0.71 | 0.71 | 3.41 | 3.41 | 6.38 | 3.43 | 3.43 | 2.69 | 2.69 | 7.23 | 7.23 | 16.76 |
| | 0.8 | 1.11(3)* | 1.11 | 0.57 | 0.57 | 5.97 | 5.97 | 7.03 | 2.40 | 2.40 | 2.04 | 2.04 | 9.38 | 9.38 | 24.86 |
| | 1.0 | 1.37(5)* | 1.37 | 1.95 | 1.95 | 4.02 | 4.02 | 2.20 | 3.43 | 3.43 | 4.94 | 4.94 | 7.55 | 7.55 | 11.18 |
| | 1.2 | 2.18(8)* | 2.18 | 0.00 | 0.00 | 6.66 | 6.66 | 39.52 | 3.21 | 3.21 | 0.00 | 0.00 | 7.49 | 7.49 | 46.97 |
| | 1.4 | (10)* | | | | | | | | | | | | | |
| | 1.6 | (10)* | | | | | | | | | | | | | |
| | 1.8 | (10)* | | | | | | | | | | | | | |
| 20 | 0.2 | 1.25(8)* | 1.25 | 2.08 | 2.08 | 4.62 | 4.62 | 3.66 | 1.54 | 1.54 | 4.15 | 4.15 | 5.40 | 5.40 | 5.56 |
| | 0.4 | 0.66(8)* | 0.66 | 0.69 | 0.69 | 4.20 | 4.20 | 12.46 | 1.01 | 1.01 | 1.38 | 1.38 | 5.21 | 5.21 | 16.18 |

| n1 | n2/n1 | Case 4 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | |
| | | Average | | | | | | | Maximum | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 1.38 | 1.38 | 0.00 | 0.00 | 6.80 | 6.80 | 1.14 | 4.80 | 4.80 | 0.00 | 0.00 | 9.76 | 9.76 | 5.10 |
| | 0.4 | 1.01 | 1.01 | 0.00 | 0.00 | 5.65 | 5.65 | 3.98 | 2.69 | 2.69 | 0.00 | 0.00 | 8.53 | 8.53 | 8.63 |
| | 0.6 | 1.67 | 1.67 | 0.00 | 0.00 | 6.12 | 6.12 | 4.34 | 3.77 | 3.77 | 0.00 | 0.00 | 9.54 | 9.54 | 8.18 |
| | 0.8 | 1.25(2)* | 1.25 | 0.00 | 0.00 | 5.36 | 5.36 | 5.64 | 3.47 | 3.47 | 0.00 | 0.00 | 8.50 | 8.50 | 14.51 |
| | 1.0 | 1.92(2)* | 1.92 | 0.00 | 0.00 | 6.03 | 6.03 | 4.75 | 4.53 | 4.53 | 0.00 | 0.00 | 8.88 | 8.88 | 12.35 |
| | 1.2 | 0.37(1)* | 0.37 | 0.00 | 0.00 | 4.40 | 4.40 | 8.32 | 2.24 | 2.24 | 0.00 | 0.00 | 7.43 | 7.43 | 14.71 |
| | 1.4 | 0.90(1)* | 0.90 | 0.00 | 0.00 | 4.65 | 4.65 | 5.09 | 3.19 | 3.19 | 0.00 | 0.00 | 6.73 | 6.73 | 17.51 |
| | 1.6 | 1.50(3)* | 1.50 | 0.00 | 0.00 | 3.89 | 3.89 | 8.75 | 3.46 | 3.46 | 0.00 | 0.00 | 7.27 | 7.27 | 24.04 |
| | 1.8 | 0.25 | 0.25 | 0.00 | 0.00 | 3.85 | 3.85 | 4.85 | 1.49 | 1.49 | 0.00 | 0.00 | 6.74 | 6.74 | 10.42 |
| 20 | 0.2 | 1.03(3)* | 1.03 | 0.00 | 0.00 | 2.76 | 2.76 | 1.65 | 2.47 | 2.47 | 0.00 | 0.00 | 4.94 | 4.94 | 4.91 |
| | 0.4 | 0.95(5)* | 0.95 | 0.00 | 0.00 | 2.77 | 2.77 | 1.86 | 2.27 | 2.27 | 0.00 | 0.00 | 4.17 | 4.17 | 3.73 |

| n1 | n2/n1 | Case 5 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | | $(Heuristic - OPT)/OPT \times 100$ | | | | | | |
| | | Average | | | | | | | Maximum | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 14.14 | 14.14 | 1.03 | 1.03 | 7.33 | 7.33 | 3.73 | 29.92 | 29.92 | 3.31 | 3.31 | 17.91 | 17.91 | 13.75 |
| | 0.4 | 12.49 | 12.95 | 1.25 | 2.69 | 4.74 | 5.01 | 9.18 | 26.62 | 27.70 | 4.55 | 7.14 | 17.53 | 17.53 | 28.38 |
| | 0.6 | 4.30(8)* | 7.47 | 1.13 | 2.16 | 0.54 | 2.16 | 2.16 | 4.76 | 7.74 | 1.79 | 4.33 | 0.60 | 4.33 | 4.33 |
| | 0.8 | (10)* | | | | | | | | | | | | | |
| | 1.0 | (10)* | | | | | | | | | | | | | |
| | 1.2 | (10)* | | | | | | | | | | | | | |
| | 1.4 | (10)* | | | | | | | | | | | | | |
| | 1.6 | (10)* | | | | | | | | | | | | | |
| | 1.8 | (10)* | | | | | | | | | | | | | |
| 20 | 0.2 | (10)* | | | | | | | | | | | | | |
| | 0.4 | (10)* | | | | | | | | | | | | | |

Since our exact solution methods are not capable to solve problems for all combinations of type-1 and type-2 jobs in a given time limit, we investigate the performance of the heuristic by examining average and maximum percent deviations from the optimal solution and global lower bound. We propose two types of heuristic algorithms, sequencing-rules based and branch-and-bound based, to solve our problem. The notation "$S_i$" in Table 6.3 and Table 6.4 symbolize the sequencing rules, which best of them determines the result of sequencing-rules based heuristic, and the notation "$B\&B$" on Table 6.3 and Table 6.4 symbolize branch-and-bound based heuristic. As can be seen from Table 6.4, we report the result of the sequencing rules one by one to be able to analyze which sequencing rule especially determines the sequencing-rules based heuristic. The sequencing rule *C*, which is denoted by, "$S_3$" is superior to other rules under all conditions. For example, when *n1*=10, *n1*/*n2*=0.4 and dominance case=1, average deviations from optimal solution are 5.14%, 7.43%, 0.69%, 3.70%, 4.30% and 7.03% for sequencing rules *A*, *B*, *C*, *D*, *E*, *F*, respectively. We also reveal that in general the sequencing-rules based heuristic algorithm gives better results than the branch-and-bound based heuristic algorithm. For example, when *n1*=10, *n1*/*n2*=0.2 and dominance case=3, average deviations from optimal solution are 1.21% and 3.67% for the best result through sequencing rule *A* to rule *F* and branch-and-bound based heuristic, respectively.

Table 6.5 shows the summary of the performance of heuristic algorithms under all possible combinations of design parameters. The results in Table 6.5 reveal similar consequences with Table 6.4. We again observe that the sequencing rule *C* is superior to other rules under all conditions and the sequencing-rules based heuristic gives better result than the branch-and-bound based heuristic. Besides, we detect that there are some combinations where the sequencing rule *C* and the branch-and-bound based heuristic performs better. For example, when we execute our heuristic on a system which common machine dominates the other machines in the system, case 2, average and maximum percentage deviation from the global lower bound is too small.

We also try to investigate the effect of number of type-1 and type-2 jobs on the heuristic algorithms performance. As can be seen from Table 6.5, the numbers of type-1 and type-2 jobs have no specific and significant effect on the performance of heuristic algorithms. For example, when *n1*=50, *n2/n1*=1.0 and dominance case= 1, average deviations from the global lower bound are 0.00% and 0.57% for the best result through sequencing rule *A* to rule *F* and branch-and-bound based heuristic, respectively. However, a little improvement is seen on the performance of the sequencing-rules based heuristic when number of type-1 jobs increase under dominance case 5. We do not observe similar results for the branch-and-bound based heuristic.

It is also observed from Table 6.5 that the ratio between number of type-1 and type-2 jobs has no significant effect on the performance of the sequencing-rules based heuristic algorithm. For example, when *n1*=30 and dominance case=3, average deviations from the global lower bound are 0.77% and 0.76% for the best result through sequencing rule *A* to rule *F* for *n1/n2*=0.4 and 2.0, respectively.

Table 6.5 reveals the effect of the related ratio on the branch-and-bound based heuristic. Only for dominance case 3, we observed that as the ratio increasing the performance of the heuristic decreases. For example, when *n1*=20 and dominance case=3, average deviations from the global lower bound are 2.24% and 16.97% for *n1/n2*=0.2 and 1.4, respectively.

As can be observed from Table 6.5, the sequencing-rules based heuristic algorithm performances are satisfactory for all problem combinations. It is capable to solve up to 150 jobs with the overall average deviation less than 5.30% and the overall maximum deviation less than 13.1%. These results are the worst-case performances; better results are obtained in the computational analyses.

As can be seen from Table 6.6, the sequencing rule *C* is very effective in solving $F3 \mid k_1 = 3, k_2 = 1, k_3 = 1, T = 2 \mid C_{\max}$ problem. The sequencing-rules based heuristic finds optimal solution quite often by using the sequencing rule *C*. Especially, in dominance case 4, our related heuristic finds optimal solutions for every comparable instances.

**Table 6.5: Heuristic Algorithms Performance–% Deviation from Global Lower Bound**

| n1 | n2/n1 | Case 1 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\left(Heuristic - GLB\right)/GLB \times 100$ | | | | | | | $\left(Heuristic - GLB\right)/GLB \times 100$ | | | | | | |
| | | Average | | | | | | | Maximum | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 8.81 | 8.81 | 2.11 | 2.11 | 9.28 | 9.28 | 6.74 | 24.19 | 24.19 | 10.48 | 10.48 | 24.64 | 24.64 | 16.91 |
| | 0.4 | 6.03 | 8.31 | 2.03 | 4.71 | 5.09 | 7.54 | 12.34 | 21.17 | 21.17 | 11.68 | 11.68 | 26.57 | 26.57 | 38.69 |
| | 0.6 | 0.57 | 4.46 | 0.50 | 3.20 | 1.52 | 4.22 | 5.93 | 5.11 | 8.02 | 4.38 | 6.79 | 14.60 | 14.60 | 27.74 |
| | 0.8 | 1.55 | 6.98 | 1.06 | 4.16 | 1.69 | 4.72 | 6.35 | 15.49 | 15.49 | 10.56 | 10.56 | 16.20 | 16.20 | 34.51 |
| | 1.0 | 0.00 | 6.19 | 0.00 | 3.71 | 0.00 | 3.75 | 3.19 | 0.00 | 8.37 | 0.00 | 6.67 | 0.00 | 6.15 | 6.67 |
| | 1.2 | 0.00 | 3.60 | 0.00 | 2.74 | 0.00 | 2.74 | 3.78 | 0.00 | 6.50 | 0.00 | 5.38 | 0.00 | 5.38 | 8.33 |
| | 1.4 | 0.00 | 5.47 | 0.00 | 4.06 | 0.00 | 3.86 | 10.42 | 0.00 | 7.31 | 0.00 | 6.07 | 0.00 | 5.67 | 24.62 |
| | 1.6 | 0.00 | 4.61 | 0.00 | 3.36 | 0.00 | 3.57 | 1.83 | 0.00 | 7.11 | 0.00 | 4.79 | 0.00 | 4.79 | 4.14 |
| | 1.8 | 0.00 | 3.19 | 0.00 | 2.17 | 0.00 | 2.17 | 15.43 | 0.00 | 5.80 | 0.00 | 4.64 | 0.00 | 4.64 | 42.41 |
| | 2.0 | 0.06 | 4.16 | 0.06 | 3.01 | 0.06 | 3.08 | 3.04 | 0.57 | 5.71 | 0.57 | 3.87 | 0.57 | 4.29 | 9.04 |
| 20 | 0.2 | 8.98 | 9.45 | 0.40 | 1.69 | 4.42 | 5.11 | 3.46 | 16.21 | 16.21 | 1.91 | 3.16 | 12.98 | 12.98 | 18.15 |
| | 0.4 | 0.69 | 3.19 | 0.04 | 1.27 | 1.16 | 2.36 | 2.02 | 4.76 | 5.30 | 0.43 | 3.31 | 11.26 | 11.26 | 5.73 |
| | 0.6 | 0.03 | 3.04 | 0.03 | 2.15 | 0.03 | 2.12 | 2.84 | 0.29 | 5.43 | 0.29 | 5.08 | 0.29 | 4.44 | 9.34 |
| | 0.8 | 0.00 | 2.91 | 0.00 | 1.66 | 0.00 | 1.89 | 2.01 | 0.00 | 4.81 | 0.00 | 2.94 | 0.00 | 4.28 | 6.89 |
| | 1.0 | 0.00 | 2.09 | 0.00 | 0.95 | 0.00 | 0.95 | 1.29 | 0.00 | 4.39 | 0.00 | 1.75 | 0.00 | 1.75 | 3.79 |
| | 1.2 | 0.02 | 2.81 | 0.00 | 1.55 | 0.00 | 1.64 | 1.62 | 0.23 | 4.44 | 0.00 | 3.01 | 0.00 | 3.09 | 3.82 |
| | 1.4 | 0.00 | 2.37 | 0.00 | 1.18 | 0.00 | 1.21 | 20.02 | 0.00 | 3.59 | 0.00 | 1.91 | 0.00 | 1.91 | 42.05 |
| | 1.6 | 0.00 | 2.41 | 0.00 | 1.31 | 0.00 | 1.38 | 1.24 | 0.00 | 3.35 | 0.00 | 2.15 | 0.00 | 2.86 | 2.79 |
| | 1.8 | 0.00 | 1.69 | 0.00 | 1.13 | 0.00 | 1.13 | 27.62 | 0.00 | 3.07 | 0.00 | 2.57 | 0.00 | 2.57 | 68.41 |
| | 2.0 | 0.02 | 2.06 | 0.02 | 1.04 | 0.02 | 1.04 | 1.66 | 0.17 | 3.28 | 0.17 | 2.16 | 0.17 | 2.11 | 2.90 |
| 30 | 0.2 | 5.33 | 6.00 | 0.02 | 1.21 | 1.69 | 2.46 | 2.39 | 15.59 | 15.59 | 0.23 | 2.01 | 6.04 | 6.04 | 5.77 |
| | 0.4 | 0.05 | 2.25 | 0.02 | 0.78 | 0.02 | 0.78 | 1.35 | 0.23 | 3.67 | 0.23 | 2.07 | 0.23 | 2.07 | 3.72 |
| | 0.6 | 0.02 | 2.24 | 0.00 | 1.05 | 0.00 | 0.99 | 1.10 | 0.20 | 3.65 | 0.00 | 3.65 | 0.00 | 3.00 | 3.62 |
| | 0.8 | 0.00 | 2.32 | 0.00 | 0.81 | 0.00 | 0.78 | 1.02 | 0.00 | 2.84 | 0.00 | 1.57 | 0.00 | 1.41 | 2.40 |
| | 1.0 | 0.00 | 2.11 | 0.00 | 0.76 | 0.00 | 0.86 | 1.23 | 0.00 | 3.07 | 0.00 | 2.07 | 0.00 | 2.07 | 2.59 |
| | 1.2 | 0.00 | 1.77 | 0.00 | 0.88 | 0.00 | 0.98 | 1.48 | 0.00 | 2.38 | 0.00 | 1.93 | 0.00 | 2.08 | 2.83 |
| | 1.4 | 0.00 | 1.89 | 0.00 | 0.87 | 0.00 | 0.90 | 26.24 | 0.00 | 2.52 | 0.00 | 1.68 | 0.00 | 1.58 | 52.86 |
| | 1.6 | 0.00 | 1.51 | 0.00 | 0.40 | 0.00 | 0.40 | 1.07 | 0.00 | 2.27 | 0.00 | 1.11 | 0.00 | 1.11 | 2.17 |
| | 1.8 | 0.00 | 1.00 | 0.00 | 0.51 | 0.00 | 0.56 | 27.34 | 0.00 | 1.95 | 0.00 | 1.38 | 0.00 | 1.38 | 58.50 |
| | 2.0 | 0.00 | 1.34 | 0.00 | 0.66 | 0.00 | 0.64 | 0.63 | 0.00 | 1.88 | 0.00 | 1.22 | 0.00 | 1.15 | 1.50 |
| 50 | 0.2 | 1.88 | 2.57 | 0.02 | 0.46 | 1.08 | 1.49 | 1.59 | 5.00 | 5.00 | 0.16 | 1.29 | 5.51 | 5.51 | 4.94 |
| | 0.4 | 1.00 | 2.14 | 0.03 | 0.45 | 0.18 | 0.52 | 1.35 | 9.42 | 9.42 | 0.27 | 0.86 | 1.57 | 1.57 | 2.67 |
| | 0.6 | 0.00 | 1.42 | 0.00 | 0.47 | 0.00 | 0.47 | 0.94 | 0.00 | 2.02 | 0.00 | 0.98 | 0.00 | 0.98 | 2.00 |
| | 0.8 | 0.00 | 1.19 | 0.00 | 0.50 | 0.00 | 0.49 | 0.55 | 0.00 | 1.88 | 0.00 | 1.44 | 0.00 | 1.15 | 1.00 |
| | 1.0 | 0.00 | 1.11 | 0.00 | 0.33 | 0.00 | 0.33 | 0.57 | 0.00 | 1.67 | 0.00 | 0.84 | 0.00 | 0.84 | 1.48 |
| | 1.2 | 0.00 | 1.14 | 0.00 | 0.25 | 0.00 | 0.27 | 0.74 | 0.00 | 1.70 | 0.00 | 0.74 | 0.00 | 0.98 | 1.30 |
| | 1.4 | 0.00 | 0.96 | 0.00 | 0.31 | 0.00 | 0.36 | 28.93 | 0.00 | 1.48 | 0.00 | 0.80 | 0.00 | 1.20 | 62.96 |
| | 1.6 | 0.00 | 0.93 | 0.00 | 0.23 | 0.00 | 0.23 | 0.40 | 0.00 | 1.51 | 0.00 | 0.31 | 0.00 | 0.31 | 0.72 |
| | 1.8 | 0.00 | 0.70 | 0.00 | 0.30 | 0.00 | 0.30 | 33.06 | 0.00 | 1.07 | 0.00 | 0.50 | 0.00 | 0.50 | 70.01 |
| | 2.0 | 0.00 | 0.75 | 0.00 | 0.21 | 0.00 | 0.21 | 0.54 | 0.00 | 1.17 | 0.00 | 0.66 | 0.00 | 0.66 | 1.27 |

**Table 6.5 (cont.)**

| n1 | n2/n1 | Case 2 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\left(Heuristic-GLB\right)/GLB\times 100$ | | | | | | | $\left(Heuristic-GLB\right)/GLB\times 100$ | | | | | | |
| | | Average | | | | | | | Maximum | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 0.03 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 1.11 | 0.28 | 0.28 | 0.00 | 0.00 | 0.00 | 0.00 | 2.81 |
| | 0.4 | 0.00 | 1.38 | 0.00 | 0.87 | 0.00 | 0.87 | 1.79 | 0.00 | 3.33 | 0.00 | 1.57 | 0.00 | 1.57 | 3.43 |
| | 0.6 | 0.07 | 1.45 | 0.00 | 0.73 | 0.00 | 0.73 | 1.14 | 0.73 | 3.42 | 0.00 | 1.54 | 0.00 | 1.54 | 3.53 |
| | 0.8 | 0.00 | 1.29 | 0.00 | 0.99 | 0.00 | 0.99 | 1.00 | 0.00 | 2.59 | 0.00 | 2.05 | 0.00 | 2.05 | 2.65 |
| | 1.0 | 0.00 | 1.94 | 0.00 | 1.01 | 0.00 | 1.01 | 1.13 | 0.00 | 2.94 | 0.00 | 2.25 | 0.00 | 2.25 | 2.09 |
| | 1.2 | 0.00 | 1.20 | 0.00 | 0.95 | 0.00 | 0.95 | 1.38 | 0.00 | 2.33 | 0.00 | 1.81 | 0.00 | 1.81 | 2.41 |
| | 1.4 | 0.00 | 1.57 | 0.00 | 0.85 | 0.00 | 0.85 | 0.88 | 0.00 | 2.28 | 0.00 | 1.41 | 0.00 | 1.41 | 1.87 |
| | 1.6 | 0.00 | 1.51 | 0.00 | 0.87 | 0.00 | 0.87 | 0.90 | 0.00 | 2.44 | 0.00 | 1.97 | 0.00 | 1.97 | 2.29 |
| | 1.8 | 0.00 | 1.38 | 0.00 | 1.00 | 0.00 | 1.00 | 0.67 | 0.00 | 2.28 | 0.00 | 1.63 | 0.00 | 1.63 | 1.22 |
| | 2.0 | 0.00 | 1.18 | 0.00 | 0.75 | 0.00 | 0.75 | 0.80 | 0.00 | 1.87 | 0.00 | 1.37 | 0.00 | 1.37 | 1.74 |
| 20 | 0.2 | 0.11 | 0.58 | 0.00 | 0.27 | 0.00 | 0.27 | 0.90 | 0.70 | 1.80 | 0.00 | 0.91 | 0.00 | 0.91 | 1.95 |
| | 0.4 | 0.00 | 0.93 | 0.00 | 0.47 | 0.00 | 0.47 | 0.86 | 0.00 | 1.81 | 0.00 | 1.17 | 0.00 | 1.17 | 1.81 |
| | 0.6 | 0.00 | 1.21 | 0.00 | 0.57 | 0.00 | 0.57 | 0.79 | 0.00 | 1.90 | 0.00 | 1.06 | 0.00 | 1.06 | 1.39 |
| | 0.8 | 0.00 | 0.80 | 0.00 | 0.40 | 0.00 | 0.40 | 0.65 | 0.00 | 1.57 | 0.00 | 0.70 | 0.00 | 0.70 | 1.48 |
| | 1.0 | 0.00 | 0.80 | 0.00 | 0.40 | 0.00 | 0.40 | 0.53 | 0.00 | 1.44 | 0.00 | 0.81 | 0.00 | 0.81 | 1.36 |
| | 1.2 | 0.00 | 0.67 | 0.00 | 0.26 | 0.00 | 0.26 | 0.61 | 0.00 | 1.15 | 0.00 | 0.52 | 0.00 | 0.52 | 1.27 |
| | 1.4 | 0.00 | 0.82 | 0.00 | 0.34 | 0.00 | 0.34 | 0.97 | 0.00 | 1.23 | 0.00 | 0.66 | 0.00 | 0.66 | 2.67 |
| | 1.6 | 0.00 | 0.73 | 0.00 | 0.39 | 0.00 | 0.39 | 0.61 | 0.00 | 1.18 | 0.00 | 0.66 | 0.00 | 0.66 | 1.19 |
| | 1.8 | 0.00 | 0.72 | 0.00 | 0.40 | 0.00 | 0.40 | 2.37 | 0.00 | 1.02 | 0.00 | 0.58 | 0.00 | 0.58 | 9.14 |
| | 2.0 | 0.00 | 0.70 | 0.00 | 0.28 | 0.00 | 0.28 | 0.51 | 0.00 | 1.05 | 0.00 | 0.77 | 0.00 | 0.77 | 0.99 |
| 30 | 0.2 | 0.01 | 0.65 | 0.00 | 0.35 | 0.00 | 0.35 | 0.61 | 0.09 | 1.05 | 0.00 | 0.70 | 0.00 | 0.70 | 1.12 |
| | 0.4 | 0.00 | 0.65 | 0.00 | 0.27 | 0.00 | 0.27 | 0.64 | 0.00 | 1.43 | 0.00 | 0.53 | 0.00 | 0.53 | 1.29 |
| | 0.6 | 0.00 | 0.81 | 0.00 | 0.32 | 0.00 | 0.32 | 0.27 | 0.00 | 1.23 | 0.00 | 0.66 | 0.00 | 0.66 | 0.84 |
| | 0.8 | 0.00 | 0.71 | 0.00 | 0.23 | 0.00 | 0.23 | 0.47 | 0.00 | 1.15 | 0.00 | 0.60 | 0.00 | 0.60 | 0.83 |
| | 1.0 | 0.00 | 0.62 | 0.00 | 0.26 | 0.00 | 0.26 | 0.50 | 0.00 | 0.89 | 0.00 | 0.43 | 0.00 | 0.43 | 0.94 |
| | 1.2 | 0.00 | 0.57 | 0.00 | 0.23 | 0.00 | 0.23 | 0.34 | 0.00 | 0.82 | 0.00 | 0.38 | 0.00 | 0.38 | 0.90 |
| | 1.4 | 0.00 | 0.49 | 0.00 | 0.18 | 0.00 | 0.18 | 4.33 | 0.00 | 0.77 | 0.00 | 0.36 | 0.00 | 0.36 | 11.27 |
| | 1.6 | 0.00 | 0.43 | 0.00 | 0.19 | 0.00 | 0.19 | 0.49 | 0.00 | 0.72 | 0.00 | 0.34 | 0.00 | 0.34 | 0.81 |
| | 1.8 | 0.00 | 0.47 | 0.00 | 0.20 | 0.00 | 0.20 | 5.23 | 0.00 | 0.69 | 0.00 | 0.40 | 0.00 | 0.40 | 14.02 |
| | 2.0 | 0.00 | 0.29 | 0.00 | 0.14 | 0.00 | 0.14 | 0.26 | 0.00 | 0.67 | 0.00 | 0.25 | 0.00 | 0.25 | 0.62 |
| 50 | 0.2 | 0.01 | 0.49 | 0.00 | 0.09 | 0.00 | 0.09 | 0.34 | 0.05 | 1.02 | 0.00 | 0.22 | 0.00 | 0.22 | 0.94 |
| | 0.4 | 0.00 | 0.55 | 0.00 | 0.12 | 0.00 | 0.12 | 0.33 | 0.00 | 0.89 | 0.00 | 0.23 | 0.00 | 0.23 | 0.62 |
| | 0.6 | 0.00 | 0.53 | 0.00 | 0.10 | 0.00 | 0.10 | 0.31 | 0.00 | 0.76 | 0.00 | 0.20 | 0.00 | 0.20 | 0.74 |
| | 0.8 | 0.00 | 0.34 | 0.00 | 0.13 | 0.00 | 0.13 | 0.16 | 0.00 | 0.67 | 0.00 | 0.28 | 0.00 | 0.28 | 0.47 |
| | 1.0 | 0.00 | 0.21 | 0.00 | 0.09 | 0.00 | 0.09 | 0.34 | 0.00 | 0.61 | 0.00 | 0.22 | 0.00 | 0.22 | 0.61 |
| | 1.2 | 0.00 | 0.30 | 0.00 | 0.10 | 0.00 | 0.10 | 0.14 | 0.00 | 0.51 | 0.00 | 0.15 | 0.00 | 0.15 | 0.57 |
| | 1.4 | 0.00 | 0.27 | 0.00 | 0.09 | 0.00 | 0.09 | 7.24 | 0.00 | 0.44 | 0.00 | 0.21 | 0.00 | 0.21 | 17.09 |
| | 1.6 | 0.00 | 0.26 | 0.00 | 0.08 | 0.00 | 0.08 | 0.18 | 0.00 | 0.47 | 0.00 | 0.18 | 0.00 | 0.18 | 0.40 |
| | 1.8 | 0.00 | 0.18 | 0.00 | 0.09 | 0.00 | 0.09 | 7.79 | 0.00 | 0.33 | 0.00 | 0.14 | 0.00 | 0.14 | 17.71 |
| | 2.0 | 0.00 | 0.21 | 0.00 | 0.06 | 0.00 | 0.06 | 0.18 | 0.00 | 0.41 | 0.00 | 0.20 | 0.00 | 0.20 | 0.38 |

**Table 6.5 (cont.)**

| n1 | n2/n1 | Case 3 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\left(Heuristic-GLB\right)/GLB\times 100$ | | | | | | | $\left(Heuristic-GLB\right)/GLB\times 100$ | | | | | | |
| | | **Average** | | | | | | | **Maximum** | | | | | | |
| | | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **B&B** | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **B&B** |
| | **0.2** | 2.08 | 2.08 | 1.93 | 1.93 | 6.80 | 6.80 | 4.42 | 4.64 | 4.64 | 6.09 | 6.09 | 9.50 | 9.50 | 11.23 |
| | **0.4** | 3.15 | 3.15 | 2.66 | 2.66 | 5.18 | 5.18 | 6.24 | 5.38 | 5.38 | 8.06 | 8.06 | 10.47 | 10.47 | 18.73 |
| | **0.6** | 2.83 | 2.83 | 2.16 | 2.16 | 5.64 | 5.64 | 9.52 | 4.62 | 4.62 | 7.17 | 7.17 | 8.80 | 8.80 | 27.54 |
| | **0.8** | 2.22 | 2.22 | 2.11 | 2.11 | 7.61 | 7.61 | 10.35 | 4.05 | 4.05 | 5.52 | 5.52 | 11.27 | 11.27 | 26.27 |
| **10** | **1.0** | 2.53 | 2.53 | 2.55 | 2.55 | 5.37 | 5.37 | 9.50 | 4.73 | 4.73 | 5.25 | 5.25 | 9.20 | 9.20 | 37.32 |
| | **1.2** | 3.33 | 3.33 | 1.63 | 1.63 | 7.51 | 7.51 | 17.98 | 4.76 | 4.76 | 5.33 | 5.33 | 12.38 | 12.38 | 48.26 |
| | **1.4** | 3.26 | 3.26 | 1.69 | 1.69 | 6.87 | 6.87 | 23.86 | 5.23 | 5.23 | 3.14 | 3.14 | 10.72 | 10.72 | 49.38 |
| | **1.6** | 2.20 | 4.37 | 1.47 | 3.17 | 7.10 | 8.04 | 27.83 | 4.57 | 12.10 | 5.93 | 7.96 | 13.69 | 13.69 | 65.54 |
| | **1.8** | 2.43 | 5.33 | 2.01 | 3.79 | 6.71 | 7.95 | 35.47 | 5.70 | 16.44 | 4.36 | 9.23 | 10.46 | 10.46 | 51.00 |
| | **2.0** | 1.97 | 6.09 | 1.26 | 4.13 | 4.08 | 6.57 | 22.29 | 6.18 | 15.93 | 3.24 | 7.67 | 10.00 | 11.50 | 61.72 |
| | **0.2** | 1.54 | 1.54 | 1.22 | 1.22 | 4.27 | 4.27 | 2.24 | 2.91 | 2.91 | 4.15 | 4.15 | 5.81 | 5.81 | 7.93 |
| | **0.4** | 1.34 | 1.34 | 1.53 | 1.53 | 5.72 | 5.72 | 5.99 | 2.46 | 2.46 | 3.23 | 3.23 | 9.69 | 9.69 | 17.03 |
| | **0.6** | 0.97 | 0.97 | 0.96 | 0.96 | 5.21 | 5.21 | 5.02 | 2.37 | 2.37 | 2.28 | 2.28 | 8.80 | 8.80 | 22.91 |
| | **0.8** | 2.06 | 2.06 | 1.41 | 1.41 | 4.58 | 4.58 | 14.41 | 2.97 | 2.97 | 3.26 | 3.26 | 6.17 | 6.17 | 29.55 |
| **20** | **1.0** | 1.67 | 1.67 | 1.28 | 1.28 | 3.95 | 3.95 | 19.92 | 2.88 | 2.88 | 2.58 | 2.58 | 5.56 | 5.56 | 36.41 |
| | **1.2** | 2.05 | 2.05 | 1.32 | 1.32 | 3.35 | 3.35 | 14.72 | 2.82 | 2.82 | 2.39 | 2.39 | 6.90 | 6.90 | 42.88 |
| | **1.4** | 1.40 | 1.40 | 1.30 | 1.30 | 4.77 | 4.77 | 16.97 | 2.53 | 2.53 | 3.22 | 3.22 | 11.16 | 11.16 | 41.42 |
| | **1.6** | 1.04 | 1.60 | 0.53 | 0.59 | 4.00 | 4.00 | 24.95 | 2.64 | 6.72 | 1.61 | 2.08 | 6.53 | 6.53 | 47.06 |
| | **1.8** | 1.15 | 4.07 | 0.91 | 2.13 | 3.46 | 4.00 | 36.25 | 2.70 | 10.94 | 2.66 | 6.51 | 5.87 | 5.87 | 64.17 |
| | **2.0** | 1.56 | 7.71 | 0.51 | 3.15 | 2.45 | 7.07 | 41.99 | 3.88 | 10.95 | 2.09 | 4.78 | 4.35 | 11.92 | 66.10 |
| | **0.2** | 0.93 | 0.93 | 0.83 | 0.83 | 2.22 | 2.22 | 3.16 | 1.82 | 1.82 | 2.16 | 2.16 | 4.21 | 4.21 | 7.21 |
| | **0.4** | 1.02 | 1.02 | 0.77 | 0.77 | 3.25 | 3.25 | 3.02 | 1.88 | 1.88 | 2.10 | 2.10 | 6.01 | 6.01 | 13.40 |
| | **0.6** | 1.45 | 1.45 | 0.61 | 0.61 | 2.89 | 2.89 | 12.06 | 2.02 | 2.02 | 1.70 | 1.70 | 5.41 | 5.41 | 21.88 |
| | **0.8** | 0.84 | 0.84 | 0.74 | 0.74 | 4.40 | 4.40 | 6.90 | 2.04 | 2.04 | 2.28 | 2.28 | 7.75 | 7.75 | 23.23 |
| **30** | **1.0** | 0.95 | 0.95 | 0.66 | 0.66 | 3.09 | 3.09 | 19.48 | 1.91 | 1.91 | 1.81 | 1.81 | 5.54 | 5.54 | 35.11 |
| | **1.2** | 0.86 | 0.86 | 0.72 | 0.72 | 3.14 | 3.14 | 15.57 | 1.74 | 1.74 | 1.53 | 1.53 | 6.09 | 6.09 | 47.34 |
| | **1.4** | 0.64 | 0.64 | 0.34 | 0.34 | 3.53 | 3.53 | 23.53 | 1.80 | 1.80 | 1.24 | 1.24 | 6.50 | 6.50 | 50.77 |
| | **1.6** | 0.80 | 1.19 | 0.91 | 1.12 | 4.26 | 4.43 | 22.42 | 1.96 | 4.82 | 1.72 | 3.07 | 9.12 | 9.12 | 58.26 |
| | **1.8** | 0.64 | 2.94 | 0.54 | 1.37 | 2.84 | 3.20 | 40.88 | 1.38 | 6.46 | 1.87 | 3.74 | 6.31 | 6.31 | 57.41 |
| | **2.0** | 1.45 | 8.59 | 0.76 | 2.71 | 2.20 | 6.35 | 31.55 | 3.68 | 11.47 | 1.99 | 5.84 | 5.09 | 10.80 | 61.23 |
| | **0.2** | 0.63 | 0.63 | 0.48 | 0.48 | 2.31 | 2.31 | 4.10 | 1.08 | 1.08 | 1.41 | 1.41 | 4.26 | 4.26 | 7.15 |
| | **0.4** | 0.66 | 0.66 | 0.51 | 0.51 | 1.74 | 1.74 | 2.88 | 1.17 | 1.17 | 0.97 | 0.97 | 3.93 | 3.93 | 13.79 |
| | **0.6** | 0.55 | 0.55 | 0.28 | 0.28 | 2.38 | 2.38 | 9.85 | 0.98 | 0.98 | 1.04 | 1.04 | 5.16 | 5.16 | 21.30 |
| | **0.8** | 0.46 | 0.46 | 0.41 | 0.41 | 2.53 | 2.53 | 8.02 | 0.98 | 0.98 | 1.35 | 1.35 | 5.41 | 5.41 | 26.86 |
| **50** | **1.0** | 0.57 | 0.57 | 0.26 | 0.26 | 2.57 | 2.57 | 21.06 | 1.09 | 1.09 | 0.70 | 0.70 | 4.18 | 4.18 | 37.92 |
| | **1.2** | 0.66 | 0.66 | 0.33 | 0.33 | 2.48 | 2.48 | 13.38 | 1.22 | 1.22 | 0.83 | 0.83 | 5.56 | 5.56 | 45.83 |
| | **1.4** | 0.80 | 0.80 | 0.40 | 0.40 | 1.81 | 1.81 | 25.07 | 1.09 | 1.09 | 1.22 | 1.22 | 4.61 | 4.61 | 53.41 |
| | **1.6** | 0.53 | 0.84 | 0.27 | 0.39 | 2.33 | 2.59 | 31.56 | 1.13 | 3.16 | 0.97 | 1.20 | 3.63 | 3.63 | 60.33 |
| | **1.8** | 0.71 | 2.90 | 0.52 | 0.57 | 2.24 | 2.43 | 42.51 | 1.24 | 7.50 | 1.25 | 1.25 | 3.94 | 4.01 | 61.40 |
| | **2.0** | 0.62 | 5.41 | 0.17 | 1.26 | 0.91 | 3.42 | 32.77 | 3.22 | 9.48 | 1.01 | 2.61 | 3.16 | 5.80 | 65.44 |

**Table 6.5 (cont.)**

| n1 | n2/n1 | Case 4 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\left(Heuristic-GLB\right)/GLB\times 100$ | | | | | | | $\left(Heuristic-GLB\right)/GLB\times 100$ | | | | | | |
| | | **Average** | | | | | | | **Maximum** | | | | | | |
| | | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **B&B** | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **B&B** |
| **10** | **0.2** | 2.32 | 2.32 | 0.92 | 0.92 | 7.79 | 7.79 | 2.08 | 5.26 | 5.26 | 3.16 | 3.16 | 10.77 | 10.77 | 8.42 |
| | **0.4** | 2.34 | 2.34 | 1.31 | 1.31 | 7.04 | 7.04 | 5.34 | 5.28 | 5.28 | 3.17 | 3.17 | 11.97 | 11.97 | 10.03 |
| | **0.6** | 2.94 | 2.94 | 1.25 | 1.25 | 7.43 | 7.43 | 5.65 | 4.76 | 4.76 | 3.58 | 3.58 | 11.47 | 11.47 | 10.75 |
| | **0.8** | 2.50 | 2.50 | 1.47 | 1.47 | 7.40 | 7.40 | 6.26 | 7.19 | 7.19 | 3.59 | 3.59 | 10.38 | 10.38 | 18.63 |
| | **1.0** | 2.96 | 2.96 | 1.39 | 1.39 | 6.50 | 6.50 | 6.79 | 4.78 | 4.78 | 3.51 | 3.51 | 9.97 | 9.97 | 14.37 |
| | **1.2** | 1.16 | 1.16 | 0.83 | 0.83 | 5.34 | 5.34 | 8.41 | 3.69 | 3.69 | 2.23 | 2.23 | 7.43 | 7.43 | 17.27 |
| | **1.4** | 2.00 | 2.00 | 1.18 | 1.18 | 5.86 | 5.86 | 5.91 | 4.21 | 4.21 | 2.48 | 2.48 | 7.54 | 7.54 | 18.41 |
| | **1.6** | 2.08 | 2.08 | 0.91 | 0.91 | 4.98 | 4.98 | 9.26 | 3.92 | 3.92 | 2.53 | 2.53 | 7.52 | 7.52 | 27.19 |
| | **1.8** | 0.90 | 0.90 | 0.65 | 0.65 | 4.52 | 4.52 | 5.54 | 1.92 | 1.92 | 1.69 | 1.69 | 7.19 | 7.19 | 12.29 |
| | **2.0** | 1.24 | 1.24 | 0.46 | 0.46 | 4.06 | 4.06 | 4.62 | 2.39 | 2.39 | 1.81 | 1.81 | 5.74 | 5.74 | 17.34 |
| **20** | **0.2** | 1.90 | 1.90 | 0.69 | 0.69 | 4.15 | 4.15 | 1.99 | 2.79 | 2.79 | 1.88 | 1.88 | 5.53 | 5.53 | 6.88 |
| | **0.4** | 1.66 | 1.66 | 0.80 | 0.80 | 3.98 | 3.98 | 3.63 | 3.11 | 3.11 | 1.48 | 1.48 | 5.10 | 5.10 | 8.72 |
| | **0.6** | 1.38 | 1.38 | 0.58 | 0.58 | 4.00 | 4.00 | 3.30 | 2.85 | 2.85 | 1.31 | 1.31 | 5.07 | 5.07 | 12.27 |
| | **0.8** | 1.20 | 1.20 | 0.40 | 0.40 | 4.55 | 4.55 | 1.90 | 2.92 | 2.92 | 0.80 | 0.80 | 5.50 | 5.50 | 4.55 |
| | **1.0** | 1.51 | 1.51 | 0.67 | 0.67 | 3.89 | 3.89 | 5.34 | 2.97 | 2.97 | 1.56 | 1.56 | 5.54 | 5.54 | 12.36 |
| | **1.2** | 1.41 | 1.41 | 0.52 | 0.52 | 3.51 | 3.51 | 4.62 | 2.43 | 2.43 | 1.23 | 1.23 | 4.51 | 4.51 | 13.95 |
| | **1.4** | 1.14 | 1.14 | 0.40 | 0.40 | 2.60 | 2.60 | 1.42 | 2.21 | 2.21 | 0.94 | 0.94 | 3.55 | 3.55 | 3.29 |
| | **1.6** | 1.30 | 1.30 | 0.46 | 0.46 | 2.85 | 2.85 | 5.35 | 2.22 | 2.22 | 1.07 | 1.07 | 4.33 | 4.33 | 17.67 |
| | **1.8** | 0.54 | 0.54 | 0.23 | 0.23 | 2.77 | 2.77 | 1.92 | 1.18 | 1.18 | 0.52 | 0.52 | 3.44 | 3.44 | 9.50 |
| | **2.0** | 0.76 | 0.76 | 0.46 | 0.46 | 2.35 | 2.35 | 6.38 | 1.53 | 1.53 | 0.91 | 0.91 | 3.75 | 3.75 | 21.60 |
| **30** | **0.2** | 0.94 | 0.94 | 0.33 | 0.33 | 2.38 | 2.38 | 0.69 | 1.82 | 1.82 | 0.62 | 0.62 | 3.35 | 3.35 | 1.77 |
| | **0.4** | 1.05 | 1.05 | 0.32 | 0.32 | 3.28 | 3.28 | 1.09 | 1.71 | 1.71 | 0.64 | 0.64 | 3.76 | 3.76 | 4.11 |
| | **0.6** | 1.01 | 1.01 | 0.40 | 0.40 | 2.78 | 2.78 | 2.71 | 1.96 | 1.96 | 0.84 | 0.84 | 3.72 | 3.72 | 7.01 |
| | **0.8** | 1.28 | 1.28 | 0.43 | 0.43 | 2.70 | 2.70 | 2.20 | 2.01 | 2.01 | 0.62 | 0.62 | 3.55 | 3.55 | 3.55 |
| | **1.0** | 0.63 | 0.63 | 0.11 | 0.11 | 2.90 | 2.90 | 0.93 | 1.90 | 1.90 | 0.53 | 0.53 | 3.74 | 3.74 | 3.38 |
| | **1.2** | 0.66 | 0.66 | 0.30 | 0.30 | 2.53 | 2.53 | 2.47 | 1.29 | 1.29 | 0.69 | 0.69 | 3.42 | 3.42 | 6.34 |
| | **1.4** | 0.82 | 0.82 | 0.37 | 0.37 | 2.33 | 2.33 | 1.29 | 1.46 | 1.46 | 0.73 | 0.73 | 3.07 | 3.07 | 3.52 |
| | **1.6** | 0.61 | 0.61 | 0.27 | 0.27 | 2.00 | 2.00 | 3.33 | 1.48 | 1.48 | 0.78 | 0.78 | 2.68 | 2.68 | 11.47 |
| | **1.8** | 0.72 | 0.72 | 0.23 | 0.23 | 1.86 | 1.86 | 2.30 | 1.35 | 1.35 | 0.65 | 0.65 | 2.67 | 2.67 | 10.74 |
| | **2.0** | 0.56 | 0.56 | 0.20 | 0.20 | 1.54 | 1.54 | 2.99 | 1.14 | 1.14 | 0.47 | 0.47 | 2.21 | 2.21 | 7.30 |
| **50** | **0.2** | 0.66 | 0.66 | 0.21 | 0.21 | 1.61 | 1.61 | 0.50 | 1.00 | 1.00 | 0.52 | 0.52 | 2.13 | 2.13 | 1.30 |
| | **0.4** | 0.48 | 0.48 | 0.19 | 0.19 | 1.83 | 1.83 | 0.67 | 0.94 | 0.94 | 0.26 | 0.26 | 2.34 | 2.34 | 1.79 |
| | **0.6** | 0.76 | 0.76 | 0.20 | 0.20 | 1.79 | 1.79 | 1.19 | 1.21 | 1.21 | 0.44 | 0.44 | 2.39 | 2.39 | 2.96 |
| | **0.8** | 0.62 | 0.62 | 0.21 | 0.21 | 1.69 | 1.69 | 1.95 | 0.97 | 0.97 | 0.38 | 0.38 | 2.44 | 2.44 | 5.94 |
| | **1.0** | 0.74 | 0.74 | 0.16 | 0.16 | 1.82 | 1.82 | 1.70 | 1.16 | 1.16 | 0.32 | 0.32 | 2.34 | 2.34 | 4.40 |
| | **1.2** | 0.66 | 0.66 | 0.18 | 0.18 | 1.36 | 1.36 | 2.46 | 0.98 | 0.98 | 0.38 | 0.38 | 1.90 | 1.90 | 6.19 |
| | **1.4** | 0.63 | 0.63 | 0.18 | 0.18 | 1.54 | 1.54 | 0.15 | 0.96 | 0.96 | 0.41 | 0.41 | 1.89 | 1.89 | 7.65 |
| | **1.6** | 0.57 | 0.57 | 0.13 | 0.13 | 1.22 | 1.22 | 2.34 | 0.88 | 0.88 | 0.27 | 0.27 | 1.66 | 1.66 | 5.79 |
| | **1.8** | 0.57 | 0.57 | 0.09 | 0.09 | 1.15 | 1.15 | 1.78 | 0.80 | 0.80 | 0.20 | 0.20 | 1.42 | 1.42 | 12.62 |
| | **2.0** | 0.24 | 0.24 | 0.09 | 0.09 | 0.95 | 0.95 | 1.76 | 0.55 | 0.55 | 0.15 | 0.15 | 1.45 | 1.45 | 3.94 |

**Table 6.5 (cont.)**

| n1 | n2/n1 | Case 5 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\left(Heuristic - GLB\right)/GLB \times 100$ | | | | | | | $\left(Heuristic - GLB\right)/GLB \times 100$ | | | | | | |
| | | Average | | | | | | | Maximum | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 18.98 | 18.98 | 5.28 | 5.28 | 11.85 | 11.85 | 8.02 | 39.52 | 39.52 | 13.04 | 13.04 | 24.64 | 24.64 | 14.91 |
| | 0.4 | 14.94 | 15.37 | 3.41 | 4.87 | 7.00 | 7.27 | 11.55 | 29.14 | 35.00 | 9.29 | 12.14 | 19.87 | 19.87 | 35.71 |
| | 0.6 | 9.02 | 10.90 | 2.80 | 3.32 | 2.93 | 4.05 | 8.00 | 19.88 | 24.22 | 12.41 | 12.41 | 12.41 | 14.91 | 31.06 |
| | 0.8 | 6.95 | 7.96 | 2.58 | 4.05 | 2.77 | 3.78 | 5.62 | 15.74 | 13.33 | 10.59 | 14.71 | 8.82 | 12.94 | 10.09 |
| | 1.0 | 3.38 | 5.27 | 0.69 | 1.49 | 0.66 | 1.92 | 4.53 | 7.28 | 10.68 | 2.71 | 5.18 | 3.63 | 7.77 | 7.28 |
| | 1.2 | 2.70 | 3.43 | 0.81 | 1.13 | 0.67 | 1.53 | 3.58 | 6.28 | 7.11 | 3.95 | 4.39 | 2.63 | 6.58 | 8.26 |
| | 1.4 | 3.12 | 5.55 | 1.40 | 3.40 | 1.40 | 3.09 | 11.38 | 5.51 | 9.51 | 4.94 | 9.51 | 4.56 | 7.06 | 25.67 |
| | 1.6 | 2.94 | 4.64 | 0.75 | 1.41 | 0.75 | 1.79 | 3.18 | 5.81 | 10.70 | 3.88 | 8.86 | 3.88 | 8.86 | 8.49 |
| | 1.8 | 1.79 | 2.84 | 1.40 | 1.96 | 1.42 | 2.44 | 15.32 | 4.64 | 6.69 | 4.64 | 6.69 | 4.55 | 7.28 | 39.79 |
| | 2.0 | 2.88 | 4.52 | 2.20 | 3.45 | 1.98 | 3.96 | 4.45 | 6.75 | 6.82 | 6.75 | 6.82 | 6.75 | 6.82 | 10.93 |
| 20 | 0.2 | 9.11 | 10.02 | 1.98 | 1.94 | 4.63 | 5.43 | 10.98 | 18.33 | 18.33 | 7.60 | 7.62 | 14.34 | 15.14 | 52.90 |
| | 0.4 | 4.13 | 5.39 | 1.17 | 1.33 | 1.08 | 1.31 | 3.32 | 7.03 | 9.79 | 6.12 | 6.12 | 7.65 | 7.65 | 6.14 |
| | 0.6 | 3.39 | 5.01 | 0.35 | 0.45 | 1.21 | 2.54 | 5.60 | 12.46 | 12.12 | 2.36 | 2.36 | 8.08 | 10.54 | 14.38 |
| | 0.8 | 2.44 | 3.20 | 0.59 | 0.66 | 0.68 | 0.99 | 2.72 | 4.41 | 4.66 | 1.89 | 1.89 | 2.16 | 2.33 | 5.41 |
| | 1.0 | 2.07 | 2.86 | 0.41 | 0.70 | 0.39 | 0.83 | 2.73 | 3.60 | 5.81 | 2.09 | 3.49 | 1.63 | 4.88 | 5.76 |
| | 1.2 | 1.37 | 2.44 | 0.18 | 0.33 | 0.14 | 0.45 | 2.48 | 4.05 | 7.34 | 1.01 | 1.01 | 0.63 | 1.52 | 4.88 |
| | 1.4 | 2.07 | 2.72 | 0.49 | 0.65 | 0.49 | 0.65 | 24.63 | 3.14 | 5.23 | 3.14 | 3.64 | 3.14 | 3.64 | 50.09 |
| | 1.6 | 1.65 | 2.61 | 0.64 | 0.60 | 0.75 | 0.79 | 2.42 | 3.78 | 4.48 | 3.78 | 3.38 | 3.78 | 3.38 | 8.55 |
| | 1.8 | 1.86 | 3.18 | 1.18 | 2.38 | 1.13 | 2.43 | 22.97 | 3.34 | 6.25 | 3.34 | 6.25 | 3.34 | 6.25 | 55.31 |
| | 2.0 | 1.98 | 2.60 | 1.31 | 2.22 | 1.34 | 2.28 | 4.37 | 3.48 | 4.18 | 3.48 | 4.18 | 3.48 | 4.18 | 6.97 |
| 30 | 0.2 | 10.44 | 12.85 | 1.07 | 2.26 | 2.76 | 3.96 | 15.21 | 24.73 | 24.20 | 6.59 | 8.31 | 6.54 | 13.75 | 42.38 |
| | 0.4 | 4.28 | 4.56 | 0.36 | 0.36 | 0.43 | 0.43 | 2.73 | 9.03 | 11.40 | 1.13 | 1.13 | 1.13 | 1.13 | 6.06 |
| | 0.6 | 2.38 | 3.49 | 0.14 | 0.22 | 0.37 | 0.70 | 3.15 | 5.15 | 8.40 | 0.79 | 0.79 | 2.92 | 5.42 | 7.25 |
| | 0.8 | 2.00 | 2.76 | 0.23 | 0.23 | 0.37 | 0.46 | 2.04 | 3.08 | 4.79 | 1.56 | 1.56 | 2.19 | 3.13 | 3.60 |
| | 1.0 | 1.65 | 3.03 | 0.27 | 0.38 | 0.37 | 0.49 | 1.70 | 3.18 | 6.53 | 1.90 | 2.85 | 1.90 | 3.01 | 2.87 |
| | 1.2 | 1.26 | 1.73 | 0.30 | 0.45 | 0.32 | 0.62 | 2.18 | 2.88 | 5.12 | 1.60 | 3.04 | 1.12 | 2.56 | 4.42 |
| | 1.4 | 1.13 | 1.98 | 0.15 | 0.34 | 0.15 | 0.35 | 26.10 | 1.78 | 3.68 | 1.47 | 2.05 | 1.47 | 2.05 | 56.46 |
| | 1.6 | 0.79 | 2.20 | 0.06 | 0.55 | 0.06 | 0.55 | 2.86 | 2.03 | 4.02 | 0.63 | 2.49 | 0.63 | 2.49 | 7.20 |
| | 1.8 | 0.92 | 1.77 | 0.59 | 1.13 | 0.59 | 1.13 | 27.65 | 2.71 | 4.53 | 2.71 | 4.53 | 2.71 | 4.53 | 58.36 |
| | 2.0 | 1.18 | 1.81 | 1.05 | 1.42 | 1.13 | 1.42 | 3.86 | 2.60 | 2.95 | 2.60 | 2.95 | 2.60 | 2.95 | 7.45 |
| 50 | 0.2 | 13.48 | 13.16 | 0.15 | 0.45 | 1.66 | 1.62 | 12.91 | 24.15 | 22.62 | 1.20 | 4.30 | 8.31 | 5.93 | 25.65 |
| | 0.4 | 2.97 | 3.52 | 0.03 | 0.03 | 0.06 | 0.06 | 10.17 | 8.28 | 9.01 | 0.29 | 0.29 | 0.59 | 0.59 | 21.22 |
| | 0.6 | 1.30 | 1.81 | 0.14 | 0.14 | 0.16 | 0.24 | 1.43 | 3.00 | 3.69 | 1.25 | 1.25 | 1.38 | 2.13 | 3.52 |
| | 0.8 | 0.98 | 1.37 | 0.07 | 0.07 | 0.20 | 0.25 | 1.19 | 1.98 | 2.97 | 0.38 | 0.38 | 0.59 | 0.99 | 3.30 |
| | 1.0 | 0.90 | 1.65 | 0.09 | 0.09 | 0.09 | 0.19 | 1.89 | 1.77 | 3.00 | 0.48 | 0.48 | 0.48 | 1.27 | 2.96 |
| | 1.2 | 0.91 | 1.35 | 0.05 | 0.05 | 0.12 | 0.13 | 1.36 | 1.64 | 2.92 | 0.19 | 0.19 | 0.58 | 0.58 | 2.60 |
| | 1.4 | 0.78 | 1.24 | 0.05 | 0.05 | 0.05 | 0.05 | 30.84 | 1.71 | 2.70 | 0.32 | 0.32 | 0.32 | 0.32 | 69.87 |
| | 1.6 | 0.74 | 1.02 | 0.04 | 0.07 | 0.07 | 0.10 | 3.05 | 1.55 | 2.13 | 0.21 | 0.57 | 0.37 | 0.57 | 7.39 |
| | 1.8 | 0.69 | 0.99 | 0.27 | 0.31 | 0.27 | 0.31 | 29.24 | 1.54 | 1.90 | 1.54 | 1.65 | 1.54 | 1.65 | 60.70 |
| | 2.0 | 0.80 | 1.30 | 0.51 | 0.82 | 0.55 | 0.85 | 3.38 | 1.67 | 2.22 | 1.67 | 2.22 | 1.67 | 2.22 | 4.52 |

**Table 6.6: Heuristic Algorithms Performance – Number of Times Optimal is Found**

| n1 | n2/n1 | Case 1 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | # of times heuristics find the optimal | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 4 | 4 | 8 | 8 | 4 | 4 | 3 |
| | 0.4 | 2(2)* | 0 | 7 | 1 | 5 | 0 | 0 |
| | 0.6 | 5(5)* | 0 | 5 | 0 | 5 | 0 | 1 |
| | 0.8 | 4(6)* | 0 | 4 | 0 | 4 | 0 | 1 |
| | 1.0 | 4(6)* | 0 | 4 | 0 | 4 | 0 | 0 |
| | 1.2 | 3(7)* | 0 | 3 | 0 | 3 | 0 | 0 |
| | 1.4 | 3(7)* | 0 | 3 | 0 | 3 | 0 | 0 |
| | 1.6 | 2(8)* | 0 | 2 | 0 | 2 | 0 | 1 |
| | 1.8 | 1(9)* | 0 | 1 | 0 | 1 | 0 | 0 |
| 20 | 0.2 | 0(9)* | 0 | 1 | 0 | 1 | 0 | 1 |
| | 0.4 | 1(8)* | 0 | 2 | 0 | 1 | 0 | 2 |

| n1 | n2/n1 | Case 2 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | # of times heuristics find the optimal | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 9 | 9 | 10 | 10 | 10 | 10 | 3 |
| | 0.4 | 8(2)* | 2 | 8 | 2 | 8 | 2 | 0 |
| | 0.6 | 5(4)* | 0 | 6 | 2 | 6 | 2 | 0 |
| | 0.8 | 5(5)* | 1 | 5 | 1 | 5 | 1 | 2 |
| | 1.0 | 5(5)* | 0 | 5 | 0 | 5 | 0 | 2 |
| | 1.2 | 2(8)* | 0 | 2 | 0 | 2 | 0 | 0 |
| | 1.4 | 3(7)* | 0 | 3 | 0 | 3 | 0 | 0 |
| | 1.6 | 2(8)* | 0 | 2 | 0 | 2 | 0 | 0 |
| | 1.8 | 3(7)* | 0 | 3 | 0 | 3 | 0 | 0 |
| 20 | 0.2 | 2(8)* | 0 | 2 | 0 | 2 | 0 | 1 |
| | 0.4 | 4(6)* | 1 | 4 | 1 | 4 | 1 | 2 |

| n1 | n2/n1 | Case 3 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | # of times heuristics find the optimal | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 3 | 3 | 6 | 6 | 0 | 0 | 2 |
| | 0.4 | 1 | 1 | 4 | 4 | 0 | 0 | 3 |
| | 0.6 | 3(5)* | 3 | 3 | 3 | 0 | 0 | 1 |
| | 0.8 | 2(3)* | 2 | 5 | 5 | 0 | 0 | 1 |
| | 1.0 | 3(5)* | 3 | 0 | 0 | 1 | 1 | 1 |
| | 1.2 | 0(8)* | 0 | 2 | 2 | 0 | 0 | 0 |
| | 1.4 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.6 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.8 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0.2 | 0(8)* | 0 | 1 | 1 | 0 | 0 | 0 |
| | 0.4 | 0(8)* | 0 | 1 | 1 | 0 | 0 | 0 |

| n1 | n2/n1 | Case 4 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | # of times heuristics find the optimal | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 4 | 4 | 10 | 10 | 0 | 0 | 6 |
| | 0.4 | 5 | 5 | 10 | 10 | 0 | 0 | 1 |
| | 0.6 | 3 | 3 | 10 | 10 | 0 | 0 | 0 |
| | 0.8 | 5(2)* | 5 | 8 | 8 | 0 | 0 | 1 |
| | 1.0 | 1(2)* | 1 | 8 | 8 | 0 | 0 | 0 |
| | 1.2 | 7(1)* | 7 | 9 | 9 | 0 | 0 | 1 |
| | 1.4 | 4(1)* | 4 | 9 | 9 | 0 | 0 | 0 |
| | 1.6 | 3(3)* | 3 | 7 | 7 | 0 | 0 | 0 |
| | 1.8 | 6 | 6 | 10 | 10 | 0 | 0 | 0 |
| 20 | 0.2 | 1(3)* | 1 | 7 | 7 | 0 | 0 | 1 |
| | 0.4 | 2(5)* | 2 | 5 | 5 | 0 | 0 | 0 |

| n1 | n2/n1 | Case 5 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | # of times heuristics find the optimal | | | | | | |
| | | S1 | S2 | S3 | S4 | S5 | S6 | B&B |
| 10 | 0.2 | 1 | 1 | 5 | 5 | 2 | 2 | 3 |
| | 0.4 | 0 | 0 | 6 | 3 | 2 | 2 | 0 |
| | 0.6 | 0(8)* | 0 | 0 | 1 | 0 | 1 | 1 |
| | 0.8 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.0 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.2 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.4 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.6 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1.8 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0.2 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0.4 | 0(10)* | 0 | 0 | 0 | 0 | 0 | 0 |

(*) The numbers in parentheses denote the number of unsolved instances out of ten within one hour time limit.

### 6.3.3 Branch-and-bound Performances

The branch-and-bound performances, i.e., CPU time, number of generated nodes and number of unsolved problems within one hour time limit are reported in Table 6.7.

We denote number of unsolved instances within one hour time limit in parentheses in Table 6.7. In calculating the average and maximum number of nodes, these unsolved instances are also included. Also, the zero CPU times indicate almost negligible computation time.

As can be seen from Table 6.7, there is an inverse relationship between the branch-and-bound algorithm's performance and the number of jobs. For example, when *n1*=10 and dominance case=1, average CPU time is 44.07 seconds and 2606.16 seconds, for *n1/n2*= 0.2 and 0.4, respectively similarly the average number of generated nodes is 5805015.60 and 4322009570 for *n1/n2*= 0.2 and 0.4, respectively. The major reason of this result is that raise in the number of jobs causes an exponential increase in the number of alternatives at each level of the tree. Another reason is that the performances of the lower bounds get worse as the number of jobs increase.

It is observed from Table 6.7 that branch-and-bound algorithm performs better for dominance case 3 and 4. For the problems which the common machine is dominated by the other machines, our algorithm can solve problem instances up to 20 jobs. Besides, if our problems are constructed for an environment where flowshop type jobs dominate parallel machine type jobs, then the branch-and-bound algorithm can solve problem instances up to 22 jobs.

**Table 6.7: Branch-and-bound Performances**

| n1 | n2/n1 | Case 1 | | | |
|---|---|---|---|---|---|
| | | CPU Time (Sec.) | | Number of Nodes | |
| | | Avg | Max | Avg | Max |
| 10 | 0.2 | 44.07 | 98.33 | 5,805,015.60 | 21,241,705.00 |
| | 0.4 | 2,606.16(2)* | 3600.00 | 432,200,957.00 | 832,710,027.00 |
| **n1** | **n2/n1** | **Case 2** | | | |
| | | CPU Time (Sec.) | | Number of Nodes | |
| | | Avg | Max | Avg | Max |
| 10 | 0.2 | 269.14 | 273.13 | 67,753,689.70 | 68,588,314.00 |
| **n1** | **n2/n1** | **Case 3** | | | |
| | | CPU Time (Sec.) | | Number of Nodes | |
| | | Avg | Max | Avg | Max |
| 10 | 0.2 | 5.10 | 5.10 | 115,115.30 | 1,150,905.00 |
| | 0.4 | 11.57 | 11.57 | 257,970.60 | 2,579,114.00 |
| | 0.6 | 2,366.34(5)* | 3600.00 | 328,805,015.20 | 858,241,905.00 |
| | 0.8 | 2,530.58(6)* | 3600.00 | 597,053,659.70 | 861,587,312.00 |
| | 1.0 | 2,426.16(5)* | 3600.00 | 577,053,659.70 | 843,247,082.00 |
| **n1** | **n2/n1** | **Case 4** | | | |
| | | CPU Time (Sec.) | | Number of Nodes | |
| | | Avg | Max | Avg | Max |
| 10 | 0.2 | 0.00 | 0.00 | 17.60 | 35.00 |
| | 0.4 | 0.00 | 0.00 | 47.90 | 102.00 |
| | 0.6 | 51.39 | 123.11 | 3,672,105.90 | 17,211,587.00 |
| | 0.8 | 1,996.57(2)* | 3600.00 | 420,865,339.00 | 824,458,622.00 |
| | 1.0 | 2,086.30(2)* | 3600.00 | 496,840,321.00 | 844,913,011.00 |
| | 1.2 | 2,926.16(7)* | 3600.00 | 578,442,431.00 | 870,513,205.00 |
| **n1** | **n2/n1** | **Case 5** | | | |
| | | CPU Time (Sec.) | | Number of Nodes | |
| | | Avg | Max | Avg | Max |
| 10 | 0.2 | 1.01 | 1.01 | 19,369.10 | 183,688.00 |
| | 0.4 | 512.08 | 2030.31 | 67,374,528.10 | 332,159,966.00 |

(*) The numbers in parentheses denote the number of unsolved instances out of ten within one hour time limit.

# CHAPTER 7

## CONCLUSION

In this thesis, we consider a scheduling problem of a manufacturing environment in which there are two manufacturing flow lines, where the third stage of the first line and the second stage of the second line are common. Each stage in the first flow line has a single machine whereas the second flow line contains two identical parallel machines in its first stage. Type-1 jobs are processed in the first flow line, whereas second flow line is dedicated to type-2 jobs. The last operation, of both types of jobs, must be processed on a common machine. The problem is to determine the sequence and schedule of all jobs at all stages of the two flow lines so that the makespan is minimized. There are many studies in scheduling literature that deal with makespan minimization problem in hybrid flowshop environment; however, scheduling in a three-stage dedicated hybrid flowshop with a common third-stage is reviewed first time in the literature with this work.

We develop a mixed integer program and a branch-and-bound algorithm with lower and upper bounding procedures to find optimal solution of the problem. Moreover, we propose two heuristic algorithms which are used as initial upper bound to our branch-and-bound algorithm and mathematical model. Besides, these heuristics provide good quality solutions at little computational effort when the computational effort to obtain an exact solution with branch-and-bound algorithm or mathematical model is prohibitive. Lower bounds are derived to evaluate the effectiveness of the proposed heuristic algorithms.

Our computational experiments are designed by generating several problem combinations using different parameters, i.e., processing time and problem size. The experiments include 5 different cases in terms of production time, and up to 150 jobs. The results are evaluated by using the following performance measures; average and maximum deviations from optimum solution and global lower bound, CPU time, the number of generated nodes and the number of unsolved problems.

We observe from our experiments that the sequencing-rules based heuristic algorithm performance is satisfactory. Its worst-case overall average deviation performance is less than 5.30% and overall maximum deviation performance is less than 13.1% for 150 jobs. The results show that for the big size problems, good solutions are obtained in reasonable time limit and mostly, sequencing rule *C* determines the result of sequencing-rules based heuristic.

The results of our computational experiments reveal that our branch-and-bound algorithm performs better if common machine is dominated by other machines or flowshop type jobs dominate parallel machine type jobs in terms of processing time. It is observed that the performance of branch-and-bound algorithm deteriorates as number of jobs increase.

There are several future extensions of this research. We suggest the following for their immediate relevance to the results achieved in this study:

(i) Development of the worst-case performance bounds for the proposed heuristics.

(ii) Extension of this studies to consider more general versions of the problem; e.g., dedicated flow lines with more than three stages.

(iii)Development of solutions by use of compu-search techniques such as Simulated Annealing, Tabu Search, or Genetic Algorithms.

(iv) Incorporation of different objectives (such as flow time and due-date related performance measures) into the problem studied in this study.

# REFERENCES

1. **ARTIBA, A.** (1994), A Rule-based Planning System for Parallel Multiproduct Manufacturing Lines, *Production Planning and Control*, 349-359, Vol. 5.

2. **HERRMANN, J. W. and LEE, C. Y.** (1992), *Three-machine Look-ahead Scheduling Problems*, Report no: 92-23, Department of Industrial and Systems Engineering, University of Florida, USA.

3. **HE, D. W., KUSIAK, A. and ARTIBA, A.** (1996), A Scheduling Problem in Glass Manufacturing, *IIE Transactions*, 129-139, Vol. 28.

4. **MEI, Q.** (1996), Scheduling Two Stage Production Lines with Multiple Machines, *Production Planning and Control,* 418-429, Vol. 7.

5. **PAUL, R. J.** (1979), A Production Scheduling Problem in the Glass-container Industry, *Operations Research*, 290-302, Vol. 22.

6. **ARTIBA, A. and RIANE, F.** (1998), An Application of a Planning and Scheduling Multi-model Approach in the Chemical Industry, *Computers in Industry*, 209-229, Vol. 36.

7. **NARASIMHAN, S. L. and PANWALKAR, S. S.** (1984), Scheduling in a Two-stage Manufacturing Process, *International Journal of Production Research*, 555-564, Vol. 22.

8. **RIANE, F., ARTIBA, A. and ELMAGHRABY, S. E.** (1998), A Hybrid Three-stage Flowshop problem: Efficient Heuristics to Minimize Makespan, *European Journal of Operations Research,* 321-329 Vol. 109.

9. **JOHNSON, S. M.** (1954), Optimal Two and Three-stage Production Schedules With Setup Times Included, *Naval Research Logistics*, 61-68, Quarter 1.

10. **OĞUZ, C., LIN, B. M. T. and CHENG, T. C. E** (1997), Two-stage Flowshop Scheduling with a Common Second-stage Machine, *Computer Opt. Res.*, 1169-1174, Vol. 24.

11. **KARP, R.M.** (1972), Reducibility Among Combinatorial Problems, *In Complexity Of Computer Computations*, R. E. Miller and J. W. Thatcher (eds.), 85-103, Plenum Press, New York.

12. **GAREY, M. R., JOHNSON, D. S. and SETHI, R. R.** (1976), The Complexity of Flowshop and Job Shop Scheduling, *Mathematics of Operations Research*, 117-129, Vol. 1.

13. **CHEN, B., GLASS, A. C., POTTS, N. C. and STRUSEVICH A. V.** (1996), A New Heuristic For Three-machine Flow Shop Scheduling, *Operations Research*, No. 6, Vol. 44.

14. **BAKER, K.R.** (1995), *Elements of Sequencing and Scheduling*, Darthmouth College, Hanover, NH

15. **BURNS, F. and ROOKER, J.** (1976), Johnson's Three-machine Flow Shop Conjecture, *Operations Research*, 578-580, Vol. 24.

16. **BURNS, F. and ROOKER, J.** (1978), Three Stage Flow Shop With Regressive Second Stage, *Operations Research*, 207-208, Vol. 26.

17. **SZWARC, W.** (1977), Optimal Two Machine Orderings in the 3xn Flow Shop Problem, *Operations Research*, 70-77, Vol. 25.

18. **SMITS, A. J. M. and BAKER, K. R** (1981), An Experimental Investigation of the Occurrence of Special Cases in the Three-machine Flowshop Problem, *International Journal of Production Research*, 737-741, Vol. 19.

19. **DUDEK, R. A. and TEUTON, O. F.** (1964), Development of M-stage decision rule for scheduling n jobs through M machines, *Operations Research*, 471-496, Vol. 12.

20. **PALMER, D. S.** (1965), Sequencing Jobs Through a Multi-stage Process in the Minimum Total Time – a Quick Method of Obtaining a Near Optimum, *Operations Research*, 101-107, Vol. 16.

21. **CAMPBELL, H. G., DUDEK, R. A., SMITH, M. L.** (1970), A Heuristic Algorithm For The n Job, m Machine Sequencing Problem, *Management Science*, 630-637, Vol. 16.

22. **PAGE, E. S.** (1961), An Approach to Scheduling Jobs on Machines, *Journal of the Royal Statistical Society*, 484–492, Vol. 23.

23. **GUPTA, J. N. D.** (1971), A Functional Heuristic Algorithm for the Flowshop Scheduling Problem, *Operations Research*, 39-48, Vol. 22.

24. **DANNENBRING, D.** (1977), An Evaluation of Flow Shop Sequencing Heuristics, *Management Science*, 1174-1182, Vol. 23.

25. **NAWAZ, M., ENSCORE Jr, E. and HAM, I.** (1983), A Heuristic Algorithm for the m-machine, n-job Flow-shop Sequencing Problem, *International Journal of Management Science*, 91-95, Vol. 11.

26. **FRAMINAN, J. M., LEISTEN, R. and RAJENDRAN, C.** (2003), Different Initial Sequences for the Heuristic of Nawaz, Enscore and Ham to Minimize Makespan, Idletime or Flowtime in the Static Permutation Flowshop Sequencing Problem, *International Journal of Production Research*, 121-148, Vol. 41.

27. **KOULAMAS, C.** (1998), A New Constructive Heuristic for the Flowshop Scheduling Problem, *European Journal of Operational Research*, 66-71, Vol. 152.

28. **SULIMAN, S.** (2000), A Two-phase Heuristic Approach to the Permutation Flow-shop Scheduling Problem, *International Journal of Production Economics*, 143-152, Vol. 64.

29. **TAILLARD, E.** (1990), Some Efficient Heuristic Methods for The Flow Shop Sequencing Problem, *European Journal of Operations Research*, 65-74, Vol. 47.

30. **PONNAMBALAM, S. G., ARAVINDAN, P. and CHANDRASEKARAN, S.** (2001), Constructive and Improvement Flow Shop Scheduling Heuristics: An Extensive Evaluation, *Production Planning and Control*, 335-344, Vol. 12.

31. **WAGNER, H. M.** (1959), An Integer Programming Model for Machine Scheduling, *Naval Research Logistics*, 131-140, Quarter 6.

32. **IGNALL, E. and SCHRAGE, L.** (1965), Application of the Branch-and-Bound technique to Some Flowshop Scheduling Problems, *Operations Research*, 400-412, Vol. 13.

33. **LOMNICKI, Z.** (1965), A Branch-and-Bound Algorithm for the Exact Solution of the Three-machine Scheduling Problem, *Operations Research*, 89-100, Vol. 16.

34. **McMAHON, G. B. and BURTON, P. G.** (1967), Flowshop Scheduling with the Branch and Bound Method, *Operations Research*, 473-481, Vol. 15.

35. **RUIZ, R. and MAROTO, C.** (2005), A Comprehensive Review and Evaluation of Permutation Flowshop Heuristics, *European Journal of Operations Research*, 479-494, Vol. 165.

36. **HEJAZI, S. R. and SAGHAFIAN, S.** (2005), Flowshop-scheduling Problems with Makespan Criterion: A Review, *International Journal of Production Research*, 2895-2929, Vol. 43.

37. **GRAHAM, R. L.** (1966), Bounds for Certain Multiprocessing Anomalies, *Bell System Technical Journal*, 1563-1581, Vol. 45.

38. **GRAHAM, R. L.** (1969), Bounds on Multiprocessing Timing Anomalies, *SIAM Journal of Applied Mathematics*, 263-269, Vol. 17.

39. **COFFMAN, E. G., Jr. and SETHI, R.** (1976), A Generalized Bound on LPT Sequencing, *RAIRO Informatique*, 17-25, Vol. 10.

40. **COFFMAN, E. G., FREDERICKSON, G. N. and LUEKER, G. S.** (1984), A Note on Expected Makespan for Largest-first Sequences of Independent Tasks on Two Processors, *Mathematics of Operations Research*, 260-266, Vol. 9.

41. **FRENK, J. B. G. and RINNOOY KAN A. H. G.** (1984), The Asymptotic Optimality of the L.P.T. Rule, *Mathematics of Operations Research*, 241-254, Vol. 12.

42. **FRENK, J. B. G. and RINNOOY KAN A. H. G.** (1986), The Rate of Convergence to optimality of the LPT rule, *Discrete Applied Mathematics*, 187-197, Vol. 14.

43. **COFFMAN, E. G., GAREY, M. R. and JOHNSON, D. S.** (1978), An Application of Bin Packing to Multiprocessor Scheduling, *SIAM Journal of Computing*, 1-17, Vol. 7.

44. **FRIESEN, D. K. and LANGSTON, M. A.** (1986), Evaluation of a Multifit-based Scheduling Algorithm, *Journal of Algorithms*, 35-59, Vol. 7.

45. **LEE, C. Y. and MASSEY, J. D.** (1988), Multiprocessor Scheduling: Combining LPT and Multifit, *Discrete Applied Mathematics*, 233-242, Vol. 105.

46. **HAOUARI, M., GHARBI, A. and JEMMALI, M.** (2006), Tight Bounds for the Identical Parallel Machine Scheduling Problem, *International Transactions in Operations Research*, 529–548, Vol. 13.

47. **ROTHKOPF, M. H.** (1966), Scheduling Independent Tasks on Parallel Processors, *Management Science*, 437–447, Vol. 12.

48. **MOKOTOFF, E.** (2002), An Exact Algorithm for the Identical Parallel Machine Scheduling Problem, *European Journal of Operational Research*, 758-769, Vol. 152.

49. **LEE, W. C., WU, C. C. and CHEN, P.** (2006), A Simulated Annealing Approach to Makespan Minimization on Identical Parallel Machines, *Int J Adv Manuf Technol*, 328-334, Vol. 31.

50. **MIN, L. and CHENG, W.** (1999), A Genetic Algorithm for Minimizing the Makespan in the Case of Scheduling Parallel Machines, *Artificial Intelligence in Engineering*, 399–403, Vol. 13.

51. **ARTHANARI, T. S. and RAMAMURTHY, K. G.** (1971), An Extension of Two Machines Sequencing Problem, *Opsearch*, 10-22, Vol. 8.

52. **GUPTA, J. N. D.** (1988), Two-stage, hybrid flowshop scheduling problem, *Operations Research*, 359-364, Vol. 38.

53. **SRISKANDARAJAH, C. and SETHI, S. P.** (1989), Scheduling Algorithms for Flexible Flowshops: Worst and Average Case Performance, *European Journal of Operations Research*, 143-160, Vol. 43.

54. **GUPTA, J. N. D. and TUNC, E. A.** (1991), Schedules for a Two-stage Hybrid Flowshop with Parallel Machines at the Second Stage, *European Journal of Operational Research,* 415-428, Vol. 77.

55. **GUPTA, J. N. D., HARIRI, A. M. A. and POTTS, C. N.** (1997), Scheduling a Two-stage Hybrid Flow Shop with Parallel Machines at the First Stage, *Annals of Operations Research*, 171–191, Vol. 69.

56. **OĞUZ, C., ERCAN, M. F., CHENG, T. C. E. and FUNG, Y. F.** (2003), Heuristic Algorithms for Multiprocessor Task Scheduling in a Two-stage Hybrid Flow-shop, *European Journal of Operational Research,* 390-403, Vol. 149.

57. **HAOUARI, M. and M'HALLAH, R.** (1997), Heuristic Algorithms for the Two-stage Hybrid Flowshop Problem, *Operations Research Letters*, 43-53, Vol. 21.

58. **WITTROCK, R. J.** (1985), Scheduling Algorithms for Flexible Flow Lines, *IBM J. Res. Develop*, 401-412, Vol. 29.

59. **WITTROCK, R. J.** (1988), An Adaptable Scheduling Algorithms for Flexible Flow Lines, *Operations Research,* 445-453, Vol. 33.

60. **BRAH, S. A. and HUNSUCKER, J. L.** (1991), Branch and Bound Algorithm for the Flow shop with Multiple Processors, *European Journal of Operational Research,* 88-99, Vol. 51.

61. **JIN, Z., YANG, Z. and ITO, T.** (2006), Metaheuristic Algorithms for the Multistage Hybrid Flowshop Scheduling problem, *Int. J. Production Economics*, 332-334, Vol. 100.

62. **RIANE, F., ARTIBA, A. and ELMAGHRABY, S. E.** (2002), Sequencing a Hybrid Two-stage Flowshop with Dedicated Machines, *International Journal of Production Research,* 4353-4380, Vol. 40.

# APPENDIX A

## NUMERICAL EXAMPLE ON THE BRANCH AND BOUND ALGORITHM

To illustrate our branch and bound algorithm, suppose that we have six jobs with the processing times given in Table A.1. The first three jobs are type-1 jobs; others are type-2 jobs.

**Table A.1: Processing Times for the Numerical Example**

| | **Processing Times** | | |
|---|---|---|---|
| **Jobs** | **Stage 1** | **Stage 2** | **Stage 3** |
| **1** | 7 | 1 | 2 |
| **2** | 3 | 3 | 6 |
| **3** | 2 | 6 | 5 |
| **4** | 1 | 0 | 8 |
| **5** | 3 | 0 | 4 |
| **6** | 7 | 0 | 4 |

Before applying the branch-and-bound algorithm on this example, we first apply the sequencing-rules based heuristic and obtain the initial upper bound for our branch and bound algorithm. Since the proposed heuristic algorithm and the global lower bound value are already calculated in Section 5.4 for the same data, we select the makespan of the sequence 4-5-6-3-2-1, which is 30, as the upper bound. In these calculations, we see that the makespan value for the schedule obtained by sequencing-rules based heuristic is equal to the global lower bound value. Thus, we can conclude that the optimal makespan value for the example is 30.

We branch from node 0.

*LEVEL 1*

The lower bounds on node 1 as follows:

$$LB_1^1 = 2 + (6+5) + \min\{3+3, 6+2\} = 19$$

$$LB_1^2 = 3 + (3+6) + \min\{3, 2\} = 14$$

$$LB_1^3 = 10 + (3+2) = 15$$

$$LB_1^4 = 2 + (8+4+4) + \min\{1, 3, 7\} = 19$$

$$LB_1^5 = \max\left\{2, \min\{0, 0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(2,0,0) - \max(2, \min(0,0))\}}{2}\right\} = 9$$

$$LB_1 = \max\{19,\ 14,\ 15,\ 19,\ 9\} = 19$$

The lower bounds on node 2:

$$LB_2^1 = 6 + (2+5) + \min\{1+7, 6+2\} = 21$$

$$LB_2^2 = 9 + (1+6) + \min\{7, 2\} = 18$$

$$LB_2^3 = 12 + (7+2) = 21$$

$$LB_2^4 = 6 + (8+4+4) + \min\{1,3,7\} = 23$$

$$LB_2^5 = \max\left\{6, \min\{0,0\}\right\} +$$
$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(6,0,0) - \max(6, \min(0,0))\}}{2}\right\} = 13$$

$$LB_2 = \max\{21,\ 18,\ 21,\ 23,\ 13\} = 23$$

The lower bounds on node 3:

$$LB_3^1 = 5 + (2+6) + \min\{1+7, 3+3\} = 19$$

$$LB_3^2 = 11 + (1+3) + \min\{7, 3\} = 18$$

$$LB_3^3 = 13 + (7+3) = 23$$

$$LB_3^4 = 5 + (8+4+4) + \min\{1,3,7\} = 22$$

$$LB_3^{\,5} = \max\left\{5, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7)-\max\{0, \max(5,0,0)-\max(5,\min(0,0))\}}{2}\right\} = 12$$

$$LB_3 = \max\{19,\ 18,\ 23,\ 22,\ 12\} = 23$$

The lower bounds on node 4 as follows:

$$LB_4^{\,1} = 8 + (2+6+5) + \min\{1+7, 3+3, 6+2\} = 27$$

$$LB_4^{\,2} = 8 + (1+3+6) + \min\{7,3,2\} = 20$$

$$LB_4^{\,3} = 8 + (7+3+2) = 20$$

$$LB_4^{\,4} = 8 + (4+4) + \min\{1,3,7\} = 17$$

$$LB_4^{\,5} = \max\left\{8, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7)-\max\{0, \max(8,0,0)-\max(8,\min(0,0))\}}{2}\right\} = 15$$

$$LB_4 = \max\{27,\ 20,\ 20,\ 17,\ 15\} = 27$$

The lower bounds on node 5 as follows:

$$LB_5^{\,1} = 4 + (2+6+5) + \min\{1+7, 3+3, 6+2\} = 23$$

$$LB_5^{\,2} = 4 + (1+3+6) + \min\{7,3,2\} = 16$$

$$LB_5^{\,3} = 4 + (7+3+2) = 16$$

$$LB_5^{\,4} = 4 + (8+4) + \min\{1,3,7\} = 17$$

$$LB_5^{\,5} = \max\left\{4, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(4,0,0) - \max(4, \min(0,0))\}}{2}\right\} = 11$$

$$LB_5 = \max\{23,\ 16,\ 16,\ 17,\ 11\} = 23$$

The lower bounds on node 6 as follows:

$$LB_6^{\,1} = 4 + (2+6+5) + \min\{1+7, 3+3, 6+2\} = 23$$

$$LB_6^{\,2} = 4 + (1+3+6) + \min\{7,3,2\} = 16$$

$$LB_6^{\,3} = 4 + (7+3+2) = 16$$

$$LB_6^{\,4} = 4 + (8+4) + \min\{1,3,7\} = 17$$

$$LB_6^{\,5} = \max\left\{4, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(4,0,0) - \max(4, \min(0,0))\}}{2}\right\} = 11$$

$LB_6 = \max\{23,\ 16,\ 16,\ 17,\ 11\} = 23$

After the calculation of the lower bounds of the jobs in set $J_u$, the node with the smallest lower bound value is selected to be branched next.

$Global\_LB\ [LEVEL(i)] = \min\left\{LB_1,\ LB_2,\ ...,\ LB_k\right\}$ where s( $J_u$ ) = k

So,

$Global\_LB\ [LEVEL(1)] = \min\{19,\ 23,\ 23,\ 27,\ 23,\ 23\} = 19$,

branch from node 1.

*LEVEL 2*

The lower bounds on node 1-2:

$LB_{1-2}^{1} = 8 + 5 + (6 + 2) = 21$

$LB_{1-2}^{2} = 11 + 6 + 2 = 19$

$LB_{1-2}^{3} = 14 + 2 = 16$

$LB_{1-2}^{4} = 8 + (8 + 4 + 4) + \min\{1, 3, 7\} = 25$

$$LB_{1-2}^{5} = \max\left\{8, \min\{0, 0\}\right\} +$$
$$\max\left\{\max\{1, 3, 7\}, \frac{(1 + 3 + 7) - \max\{0, \max(8, 0, 0) - \max(8, \min(0, 0))\}}{2}\right\} = 15$$

$LB_{1-2} = \max\{21,\ 19,\ 16,\ 25,\ 15\} = 25$

The lower bounds on node 1-3:

$$LB_{1-3}^{1} = 7 + 6 + (3 + 3) = 19$$

$$LB_{1-3}^{2} = 13 + 3 + 3 = 19$$

$$LB_{1-3}^{3} = 15 + 3 = 18$$

$$LB_{1-3}^{4} = 7 + (8 + 4 + 4) + \min\{1, 3, 7\} = 24$$

$$LB_{1-3}^{5} = \max\left\{7, \min\{0, 0\}\right\} +$$
$$\max\left\{\max\{1, 3, 7\}, \frac{(1 + 3 + 7) - \max\{0, \max(7, 0, 0) - \max(7, \min(0, 0))\}}{2}\right\} = 14$$

$$LB_{1-3} = \max\{19,\ 19,\ 18, 24,\ 14\} = 24$$

The lower bounds on node 1-4 as follows:

$$LB_{1-4}^{1} = 10 + (6 + 5) + \min\{3 + 3, 6 + 2\} = 27$$

$$LB_{1-4}^{2} = 10 + (3 + 6) + \min\{3, 2\} = 21$$

$$LB_{1-4}^{3} = 10 + (3 + 2) = 15$$

$$LB_{1-4}^{4} = 10 + (4 + 4) + \min\{1, 3, 7\} = 19$$

$$LB_{1-4}^{5} = \max\left\{10, \min\{0,0\}\right\} +$$
$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7)-\max\{0, \max(10,0,0)-\max(10,\min(0,0))\}}{2}\right\} = 17$$

$$LB_{1-4} = \max\{27,\ 21,\ 15,\ 19,\ 17\} = 27$$

The lower bounds on node 1-5 as follows:

$$LB_{1-5}^{1} = 6 + (6+5) + \min\{3+3, 6+2\} = 23$$

$$LB_{1-5}^{2} = 6 + (3+6) + \min\{3,2\} = 17$$

$$LB_{1-5}^{3} = 6 + (3+2) = 11$$

$$LB_{1-5}^{4} = 6 + (8+4) + \min\{1,3,7\} = 19$$

$$LB_{1-5}^{5} = \max\left\{6, \min\{0,0\}\right\} +$$
$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7)-\max\{0, \max(6,0,0)-\max(6,\min(0,0))\}}{2}\right\} = 13$$

$$LB_{1-5} = \max\{23,\ 17,\ 11,\ 19,\ 13\} = 23$$

The lower bounds on node 1-6 as follows:

$$LB_{1-6}^{1} = 6 + (6+5) + \min\{3+3, 6+2\} = 23$$

$$LB_{1-6}^{2} = 6 + (3+6) + \min\{3, 2\} = 17$$

$$LB_{1-6}^{3} = 6 + (3+2) = 11$$

$$LB_{1-6}^{4} = 6 + (8+4) + \min\{1, 3, 7\} = 19$$

$$LB_{1-6}^{5} = \max\left\{6, \min\{0, 0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(6,0,0) - \max(6, \min(0,0))\}}{2}\right\} = 13$$

$$LB_{1-6} = \max\{23,\ 17,\ 11,\ 19,\ 13\} = 23$$

$$Global\_LB\ [LEVEL(2)] = \min\{25,\ 24,\ 27,\ 23,\ 23\} = 23,$$

branch from node 10 or node 11 since they have the smallest lower bound value; choose one of them arbitrary.

*LEVEL 3*

The lower bounds on node 1-5-2:

$$LB_{1-5-2}^{1} = 12 + 5 + (2+6) = 25$$

$$LB_{1-5-2}^{2} = 15 + 6 + 2 = 23$$

$$LB_{1-5-2}^{3} = 18 + 2 = 20$$

$$LB_{1-5-2}^{4} = 12 + (8+4) + \min\{1,7\} = 25$$

$$LB_{1-5-2}^{5} = \max\left\{12, \min\{9,0\}\right\} +$$

$$\max\left\{\max\{1,7\}, \frac{(1+7) - \max\{0, \max(12,9,0) - \max(12, \min(9,0))\}}{2}\right\} = 19$$

$$LB_{1-5-2} = \max\{25,\ 23,\ 20,\ 25,\ 19\} = 25$$

The lower bounds on node 1-5-3 as follows:

$$LB_{1-5-3}^{1} = 11 + 6 + (3+3) = 23$$

$$LB_{1-5-3}^{2} = 17 + 3 + 3 = 23$$

$$LB_{1-5-3}^{3} = 19 + 3 = 22$$

$$LB_{1-5-3}^{4} = 11 + (8+4) + \min\{1,7\} = 24$$

$$LB_{1-5-3}^{5} = \max\left\{11, \min\{9,0\}\right\} +$$

$$\max\left\{\max\{1,7\}, \frac{(1+7) - \max\{0, \max(11,9,0) - \max(11, \min(9,0))\}}{2}\right\} = 18$$

$$LB_{1-5-3} = \max\{23,\ 23,\ 22,\ 24,\ 18\} = 24$$

The lower bounds on node 1-5-4 as follows:

$$LB_{1-5-4}^{1} = 14 + (6+5) + \min\{3+3, 6+2\} = 31$$

$$LB_{1-5-4}^{2} = 14 + (3+6) + \min\{3, 2\} = 25$$

$$LB_{1-5-4}^{3} = 14 + (3+2) = 19$$

$$LB_{1-5-4}^{4} = 14 + 4 + \min\{1, 7\} = 19$$

$$LB_{1-5-4}^{5} = \max\left\{14, \min\{9, 0\}\right\} +$$

$$\max\left\{\max\{1,7\}, \frac{(1+7) - \max\{0, \max(14,9,0) - \max(14, \min(9,0))\}}{2}\right\} = 21$$

$$LB_{1-5-4} = \max\{31,\ 25,\ 19,\ 19,\ 21\} = 31$$

The lower bounds on node 1-5-6 as follows:

$$LB_{1-5-6}^{1} = 10 + (6+5) + \min\{3+3, 6+2\} = 27$$

$$LB_{1-5-6}^{2} = 10 + (3+6) + \min\{3, 2\} = 21$$

$$LB_{1-5-6}^{3} = 10 + (3+2) = 15$$

$$LB_{1-5-6}^{4} = 10 + 8 + \min\{1, 7\} = 19$$

$$LB_{1-5-6}^{5} = \max\left\{10, \min\{9,0\}\right\} +$$
$$\max\left\{\max\{1,7\}, \frac{(1+7)-\max\{0, \max(10,9,0)-\max(10,\min(9,0))\}}{2}\right\} = 17$$

$$LB_{1-5-6} = \max\{27,\ 21,\ 15,\ 19,\ 17\} = 27$$

$Global\_LB\ [LEVEL(3)] = \min\{25,\ 24,\ 31,\ 27\} = 24$,

branch from node 13

*LEVEL 4*

The lower bounds on node 1-5-3-2 as follows:

$$LB_{1-5-3-2}^{1} = 17 + 0 + 0 = 17$$

$$LB_{1-5-3-2}^{2} = 20 + 0 + 0 = 20$$

$$LB_{1-5-3-2}^{3} = 23 + 0 = 23$$

$$LB_{1-5-3-2}^{4} = 17 + (8+4) + \min\{1,7\} = 30$$

$$LB_{1-5-3-2}^{5} = \max\left\{17, \min\{9,0\}\right\} +$$
$$\max\left\{\max\{1,7\}, \frac{(1+7)-\max\{0, \max(17,9,0)-\max(17,\min(9,0))\}}{2}\right\} = 24$$

$$LB_{1-5-3-2} = \max\{17,\ 20,\ 23,\ 30,\ 24\} = 30$$

The lower bounds on node 1-5-3-4 as follows:

$$LB_{1-5-3-4}^{1} = 19 + 6 + (3+3) = 31$$

$$LB_{1-5-3-4}^{2} = 19 + 3 + 3 = 25$$

$$LB_{1-5-3-4}^{3} = 19 + 3 = 22$$

$$LB_{1-5-3-4}^{4} = 19 + 4 + \min\{1,7\} = 24$$

$$LB_{1-5-3-4}^{5} = \max\left\{19, \min\{9,0\}\right\} +$$
$$\max\left\{\max\{1,7\}, \frac{(1+7) - \max\{0, \max(19,9,0) - \max(19, \min(9,0))\}}{2}\right\} = 26$$

$$LB_{1-5-3-4} = \max\{31,\ 25,\ 22,\ 24,\ 26\} = 31$$

The lower bounds on node 1-5-3-6 as follows:

$$LB_{1-5-3-6}^{1} = 15 + 6 + (3+3) = 27$$

$$LB_{1-5-3-6}^{2} = 15 + 3 + 3 = 21$$

$$LB_{1-5-3-6}^{3} = 15 + 3 = 18$$

$$LB_{1-5-3-6}^{4} = 15 + 8 + \min\{1,7\} = 24$$

$$LB_{1-5-3-6}^{5} = \max\left\{15, \min\{9,0\}\right\} +$$
$$\max\left\{\max\{1,7\}, \frac{(1+7)-\max\{0, \max(15,9,0)-\max(15,\min(9,0))\}}{2}\right\} = 22$$

$$LB_{1-5-3-6} = \max\{27,\ 21,\ 18,\ 24,\ 22\} = 27$$

$Global\_LB\ [LEVEL(4)] = \min\{30,\ 31,\ 27\} = 27$,

branch from node 13

*LEVEL 5*

The lower bounds on node 1-5-3-6-2 as follows:

$$LB_{1-5-3-6-2}^{1} = 21+0+0 = 21$$
$$LB_{1-5-3-6-2}^{2} = 24+0+0 = 24$$
$$LB_{1-5-3-6-2}^{3} = 27+0 = 27$$
$$LB_{1-5-3-6-2}^{4} = 21+8+1 = 30$$
$$LB_{1-5-3-6-2}^{5} = \max\left\{21, \min\{9,22\}\right\} +$$
$$\max\left\{1, \frac{1-\max\{0, \max(21,9,22)-\max(21,\min(9,22))\}}{2}\right\} = 22$$

$$LB_{1-5-3-6-2} = \max\{21,\ 24,\ 27,\ 30,\ 22\} = 30$$

The lower bounds on node 1-5-3-6-4 as follows:

$$LB_{1-5-3-6-4}^{1} = 23 + 6 + (3+3) = 35$$

$$LB_{1-5-3-6-4}^{2} = 23 + 3 + 3 = 29$$

$$LB_{1-5-3-6-4}^{3} = 23 + 3 = 26$$

$$LB_{1-5-3-6-4}^{4} = 23 + 0 + 1 = 24$$

$$LB_{1-5-3-6-4}^{5} = \max\left\{23, \min\{9,22\}\right\} +$$

$$\max\left\{1, \frac{1-\max\{0, \max(23,9,22) - \max(23, \min(9,22))\}}{2}\right\} = 24$$

$$LB_{1-5-3-6-4} = \max\{35,\ 29,\ 26,\ 24,\ 24\} = 35$$

$Global\_LB\ [LEVEL(5)] = \min\{30,\ 35\} = 30$,

branch from node 19

*LEVEL 6*

The lower bounds on node 1-5-3-6-2-4 as follows:

$$LB_{1-5-3-6-2-4}^{1} = 29 + 0 + 0 = 29$$

$$LB_{1-5-3-6-2}^{2} = 29 + 0 + 0 = 29$$

$$LB_{1-5-3-6-2-4}^{3} = 29 + 0 = 29$$

$$LB^{4}_{1-5-3-6-2-4} = 29+0+1=30$$

$$LB^{5}_{1-5-3-6-2-4} = \max\left\{29, \min\{9,22\}\right\} +$$
$$\max\left\{1, \frac{1-\max\{0, \max(29,9,22)-\max(29,\min(9,22))\}}{2}\right\} = 30$$

$$LB_{1-5-3-6-2-4} = \max\{29,\ 29,\ 29,\ 30,\ 30\} = 30$$

$$C_{\max} = 30$$

At last level of the tree, we obtain a complete schedule with the makespan value 30 that is equal to the optimal one. Hence, we cannot get better result than 30; so we can fathom all other unbranched nodes.

Since there is no unfathomed or nonbranched node, we do not backtrack and the latest updated upper bound is accepted as the solution of the algorithm, 30.

Branching scheme of the numerical example is given in Figure A.1.

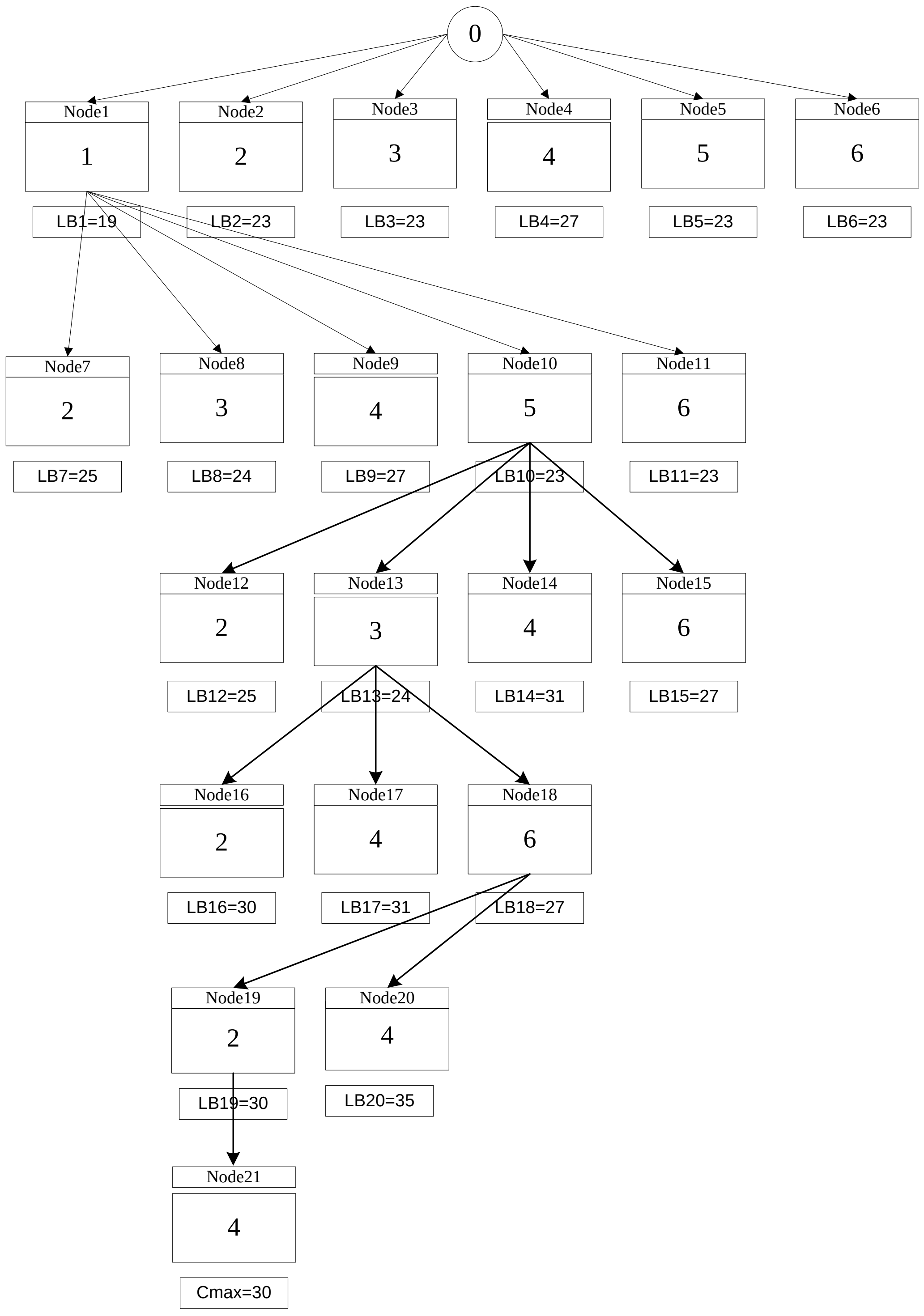

**Figure A.1: Branching Scheme of the Numerical Example**

# APPENDIX B

## NUMERICAL EXAMPLE ON THE BRANCH-AND-BOUND BASED HEURISTIC

To illustrate the proposed heuristic algorithm, suppose that we have six jobs with the processing times given in Table B.1. The first three jobs are type-1 jobs; others are type-2 jobs.

**Table B.1: Processing Times for the Numerical Example**

| | **Processing Times** | | |
|---|---|---|---|
| **Jobs** | **Stage 1** | **Stage 2** | **Stage 3** |
| **1** | 7 | 1 | 2 |
| **2** | 3 | 3 | 6 |
| **3** | 2 | 6 | 5 |
| **4** | 1 | 0 | 8 |
| **5** | 3 | 0 | 4 |
| **6** | 7 | 0 | 4 |

We branch from node 0.

*LEVEL 1*

The lower bounds on node 1 as follows:

$$LB_1^1 = 2 + (6+5) + \min\{3+3, 6+2\} = 19$$

$$LB_1^2 = 3 + (3+6) + \min\{3, 2\} = 14$$

$$LB_1^3 = 10 + (3+2) = 15$$

$$LB_1^4 = 2 + (8+4+4) + \min\{1, 3, 7\} = 19$$

$$LB_1^5 = \max\left\{2, \min\{0, 0\}\right\} +$$
$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(2,0,0) - \max(2, \min(0,0))\}}{2}\right\} = 9$$

$$LB_1 = \max\{19,\ 14,\ 15,\ 19,\ 9\} = 19$$

The lower bounds on node 2:

$$LB_2^1 = 6 + (2+5) + \min\{1+7, 6+2\} = 21$$

$$LB_2^2 = 9 + (1+6) + \min\{7, 2\} = 18$$

$$LB_2^3 = 12 + (7+2) = 21$$

$$LB_2^4 = 6 + (8+4+4) + \min\{1, 3, 7\} = 23$$

$$LB_2^{\,5} = \max\left\{6, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(6,0,0) - \max(6, \min(0,0))\}}{2}\right\} = 13$$

$$LB_2 = \max\{21,\ 18,\ 21,\ 23,\ 13\} = 23$$

The lower bounds on node 3:

$$LB_3^{\,1} = 5 + (2+6) + \min\{1+7, 3+3\} = 19$$

$$LB_3^{\,2} = 11 + (1+3) + \min\{7,3\} = 18$$

$$LB_3^{\,3} = 13 + (7+3) = 23$$

$$LB_3^{\,4} = 5 + (8+4+4) + \min\{1,3,7\} = 22$$

$$LB_3^{\,5} = \max\left\{5, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(5,0,0) - \max(5, \min(0,0))\}}{2}\right\} = 12$$

$$LB_3 = \max\{19,\ 18,\ 23,\ 22,\ 12\} = 23$$

The lower bounds on node 4 as follows:

$$LB_4^1 = 8 + (2 + 6 + 5) + \min\{1 + 7, 3 + 3, 6 + 2\} = 27$$

$$LB_4^2 = 8 + (1 + 3 + 6) + \min\{7, 3, 2\} = 20$$

$$LB_4^3 = 8 + (7 + 3 + 2) = 20$$

$$LB_4^4 = 8 + (4 + 4) + \min\{1, 3, 7\} = 17$$

$$LB_4^5 = \max\left\{8, \min\{0, 0\}\right\} +$$
$$\max\left\{\max\{1, 3, 7\}, \frac{(1 + 3 + 7) - \max\{0, \max(8, 0, 0) - \max(8, \min(0, 0))\}}{2}\right\} = 15$$

$$LB_4 = \max\{27,\ 20,\ 20,\ 17,\ 15\} = 27$$

The lower bounds on node 5 as follows:

$$LB_5^1 = 4 + (2 + 6 + 5) + \min\{1 + 7, 3 + 3, 6 + 2\} = 23$$

$$LB_5^2 = 4 + (1 + 3 + 6) + \min\{7, 3, 2\} = 16$$

$$LB_5^3 = 4 + (7 + 3 + 2) = 16$$

$$LB_5^4 = 4 + (8 + 4) + \min\{1, 3, 7\} = 17$$

$$LB_5^5 = \max\left\{4, \min\{0,0\}\right\} + \max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(4,0,0) - \max(4,\min(0,0))\}}{2}\right\} = 11$$

$$LB_5 = \max\{23,\ 16,\ 16,\ 17,\ 11\} = 23$$

The lower bounds on node 6 as follows:

$$LB_6^1 = 4 + (2+6+5) + \min\{1+7, 3+3, 6+2\} = 23$$

$$LB_6^2 = 4 + (1+3+6) + \min\{7,3,2\} = 16$$

$$LB_6^3 = 4 + (7+3+2) = 16$$

$$LB_6^4 = 4 + (8+4) + \min\{1,3,7\} = 17$$

$$LB_6^5 = \max\left\{4, \min\{0,0\}\right\} + \max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(4,0,0) - \max(4,\min(0,0))\}}{2}\right\} = 11$$

$$LB_6 = \max\{23,\ 16,\ 16,\ 17,\ 11\} = 23$$

After the calculation of the lower bounds of the jobs in set $J_u$, the node with the smallest lower bound value is selected to be branched next.

So,

$Global\_LB\ [LEVEL(1)] = \min\{19,\ 23,\ 23,\ 27,\ 23,\ 23\} = 19$,

branch from node 1.

*LEVEL 2*

The lower bounds on node 1-2:

$$LB_{1-2}^{1} = 8+5+(6+2) = 21$$

$$LB_{1-2}^{2} = 11+6+2 = 19$$

$$LB_{1-2}^{3} = 14+2 = 16$$

$$LB_{1-2}^{4} = 8+(8+4+4)+\min\{1,3,7\} = 25$$

$$LB_{1-2}^{5} = \max\left\{8, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7)-\max\{0, \max(8,0,0)-\max(8,\min(0,0))\}}{2}\right\} = 15$$

$$LB_{1-2} = \max\{21,\ 19,\ 16,\ 25,\ 15\} = 25$$

The lower bounds on node 1-3:

$$LB_{1-3}^{1} = 7+6+(3+3) = 19$$

$$LB_{1-3}^{2} = 13+3+3 = 19$$

$$LB_{1-3}^{3} = 15+3 = 18$$

$$LB_{1-3}^{4} = 7 + (8 + 4 + 4) + \min\{1,3,7\} = 24$$

$$LB_{1-3}^{5} = \max\left\{7, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(7,0,0) - \max(7,\min(0,0))\}}{2}\right\} = 14$$

$$LB_{1-3} = \max\{19,\ 19,\ 18, 24,\ 14\} = 24$$

The lower bounds on node 1-4 as follows:

$$LB_{1-4}^{1} = 10 + (6 + 5) + \min\{3+3, 6+2\} = 27$$

$$LB_{1-4}^{2} = 10 + (3 + 6) + \min\{3,2\} = 21$$

$$LB_{1-4}^{3} = 10 + (3 + 2) = 15$$

$$LB_{1-4}^{4} = 10 + (4 + 4) + \min\{1,3,7\} = 19$$

$$LB_{1-4}^{5} = \max\left\{10, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(10,0,0) - \max(10,\min(0,0))\}}{2}\right\} = 17$$

$$LB_{1-4} = \max\{27,\ 21,\ 15,\ 19,\ 17\} = 27$$

The lower bounds on node 1-5 as follows:

$$LB_{1-5}^{1} = 6 + (6+5) + \min\{3+3, 6+2\} = 23$$

$$LB_{1-5}^{2} = 6 + (3+6) + \min\{3, 2\} = 17$$

$$LB_{1-5}^{3} = 6 + (3+2) = 11$$

$$LB_{1-5}^{4} = 6 + (8+4) + \min\{1, 3, 7\} = 19$$

$$LB_{1-5}^{5} = \max\left\{6, \min\{0,0\}\right\} +$$

$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7) - \max\{0, \max(6,0,0) - \max(6, \min(0,0))\}}{2}\right\} = 13$$

$$LB_{1-5} = \max\{23,\ 17,\ 11,\ 19,\ 13\} = 23$$

The lower bounds on node 1-6 as follows:

$$LB_{1-6}^{1} = 6 + (6+5) + \min\{3+3, 6+2\} = 23$$

$$LB_{1-6}^{2} = 6 + (3+6) + \min\{3, 2\} = 17$$

$$LB_{1-6}^{3} = 6 + (3+2) = 11$$

$$LB_{1-6}^{4} = 6 + (8+4) + \min\{1, 3, 7\} = 19$$

$$LB_{1-6}^{5} = \max\left\{6, \min\{0,0\}\right\} +$$
$$\max\left\{\max\{1,3,7\}, \frac{(1+3+7)-\max\{0, \max(6,0,0)-\max(6,\min(0,0))\}}{2}\right\} = 13$$

$$LB_{1-6} = \max\{23,\ 17,\ 11,\ 19,\ 13\} = 23$$

$$Global\_LB\ [LEVEL(2)] = \min\{25,\ 24,\ 27,\ 23,\ 23\} = 23,$$

branch from node 10 or node 11 since they have the smallest lower bound value; choose one of them arbitrary.

*LEVEL 3*

The lower bounds on node 1-5-2:

$$LB_{1-5-2}^{1} = 12 + 5 + (2+6) = 25$$

$$LB_{1-5-2}^{2} = 15 + 6 + 2 = 23$$

$$LB_{1-5-2}^{3} = 18 + 2 = 20$$

$$LB_{1-5-2}^{4} = 12 + (8+4) + \min\{1,7\} = 25$$

$$LB_{1-5-2}^{5} = \max\left\{12, \min\{9,0\}\right\} +$$
$$\max\left\{\max\{1,7\}, \frac{(1+7)-\max\{0, \max(12,9,0)-\max(12,\min(9,0))\}}{2}\right\} = 19$$

$$LB_{1-5-2} = \max\{25,\ 23,\ 20,\ 25,\ 19\} = 25$$

The lower bounds on node 1-5-3 as follows:

$$LB_{1-5-3}^{1} = 11 + 6 + (3+3) = 23$$

$$LB_{1-5-3}^{2} = 17 + 3 + 3 = 23$$

$$LB_{1-5-3}^{3} = 19 + 3 = 22$$

$$LB_{1-5-3}^{4} = 11 + (8+4) + \min\{1,7\} = 24$$

$$LB_{1-5-3}^{5} = \max\left\{11, \min\{9,0\}\right\} +$$

$$\max\left\{\max\{1,7\}, \frac{(1+7) - \max\{0, \max(11,9,0) - \max(11, \min(9,0))\}}{2}\right\} = 18$$

$$LB_{1-5-3} = \max\{23,\ 23,\ 22,\ 24,\ 18\} = 24$$

The lower bounds on node 1-5-4 as follows:

$$LB_{1-5-4}^{1} = 14 + (6+5) + \min\{3+3, 6+2\} = 31$$

$$LB_{1-5-4}^{2} = 14 + (3+6) + \min\{3,2\} = 25$$

$$LB_{1-5-4}^{3} = 14 + (3+2) = 19$$

$$LB_{1-5-4}^{4} = 14 + 4 + \min\{1,7\} = 19$$

$$LB_{1-5-4}^{5} = \max\left\{14, \min\{9,0\}\right\} + \max\left\{\max\{1,7\}, \frac{(1+7) - \max\{0, \max(14,9,0) - \max(14, \min(9,0))\}}{2}\right\} = 21$$

$$LB_{1-5-4} = \max\{31,\ 25,\ 19,\ 19,\ 21\} = 31$$

The lower bounds on node 1-5-6 as follows:

$$LB_{1-5-6}^{1} = 10 + (6+5) + \min\{3+3, 6+2\} = 27$$

$$LB_{1-5-6}^{2} = 10 + (3+6) + \min\{3,2\} = 21$$

$$LB_{1-5-6}^{3} = 10 + (3+2) = 15$$

$$LB_{1-5-6}^{4} = 10 + 8 + \min\{1,7\} = 19$$

$$LB_{1-5-6}^{5} = \max\left\{10, \min\{9,0\}\right\} + \max\left\{\max\{1,7\}, \frac{(1+7) - \max\{0, \max(10,9,0) - \max(10, \min(9,0))\}}{2}\right\} = 17$$

$$LB_{1-5-6} = \max\{27,\ 21,\ 15,\ 19,\ 17\} = 27$$

$Global\_LB\ [LEVEL(3)] = \min\{25,\ 24,\ 31,\ 27\} = 24$,

branch from node 13

*LEVEL 4*

The lower bounds on node 1-5-3-2 as follows:

$$LB_{1-5-3-2}^{1} = 17 + 0 + 0 = 17$$

$$LB_{1-5-3-2}^{2} = 20 + 0 + 0 = 20$$

$$LB_{1-5-3-2}^{3} = 23 + 0 = 23$$

$$LB_{1-5-3-2}^{4} = 17 + (8 + 4) + \min\{1, 7\} = 30$$

$$LB_{1-5-3-2}^{5} = \max\left\{17, \min\{9, 0\}\right\} +$$
$$\max\left\{\max\{1, 7\}, \frac{(1 + 7) - \max\{0, \max(17, 9, 0) - \max(17, \min(9, 0))\}}{2}\right\} = 24$$

$$LB_{1-5-3-2} = \max\{17,\ 20,\ 23,\ 30,\ 24\} = 30$$

The lower bounds on node 1-5-3-4 as follows:

$$LB_{1-5-3-4}^{1} = 19 + 6 + (3 + 3) = 31$$

$$LB_{1-5-3-4}^{2} = 19 + 3 + 3 = 25$$

$$LB_{1-5-3-4}^{3} = 19 + 3 = 22$$

$$LB_{1-5-3-4}^{4} = 19 + 4 + \min\{1, 7\} = 24$$

$$LB_{1-5-3-4}^{5} = \max\left\{19, \min\{9,0\}\right\} +$$
$$\max\left\{\max\{1,7\}, \frac{(1+7)-\max\{0, \max(19,9,0)-\max(19,\min(9,0))\}}{2}\right\} = 26$$

$$LB_{1-5-3-4} = \max\{31,\ 25,\ 22,\ 24,\ 26\} = 31$$

The lower bounds on node 1-5-3-6 as follows:

$$LB_{1-5-3-6}^{1} = 15+6+(3+3) = 27$$

$$LB_{1-5-3-6}^{2} = 15+3+3 = 21$$

$$LB_{1-5-3-6}^{3} = 15+3 = 18$$

$$LB_{1-5-3-6}^{4} = 15+8+\min\{1,7\} = 24$$

$$LB_{1-5-3-6}^{5} = \max\left\{15, \min\{9,0\}\right\} +$$
$$\max\left\{\max\{1,7\}, \frac{(1+7)-\max\{0, \max(15,9,0)-\max(15,\min(9,0))\}}{2}\right\} = 22$$

$$LB_{1-5-3-6} = \max\{27,\ 21,\ 18,\ 24,\ 22\} = 27$$

$$Global\_LB\ [LEVEL(4)] = \min\{30,\ 31,\ 27\} = 27,$$

branch from node 13

*LEVEL 5*

The lower bounds on node 1-5-3-6-2 as follows:

$$LB_{1-5-3-6-2}^{1} = 21 + 0 + 0 = 21$$

$$LB_{1-5-3-6-2}^{2} = 24 + 0 + 0 = 24$$

$$LB_{1-5-3-6-2}^{3} = 27 + 0 = 27$$

$$LB_{1-5-3-6-2}^{4} = 21 + 8 + 1 = 30$$

$$LB_{1-5-3-6-2}^{5} = \max\left\{21, \min\{9, 22\}\right\} +$$
$$\max\left\{1, \frac{1 - \max\{0, \max(21, 9, 22) - \max(21, \min(9, 22))\}}{2}\right\} = 22$$

$$LB_{1-5-3-6-2} = \max\{21,\ 24,\ 27,\ 30,\ 22\} = 30$$

The lower bounds on node 1-5-3-6-4 as follows:

$$LB_{1-5-3-6-4}^{1} = 23 + 6 + (3 + 3) = 35$$

$$LB_{1-5-3-6-4}^{2} = 23 + 3 + 3 = 29$$

$$LB_{1-5-3-6-4}^{3} = 23 + 3 = 26$$

$$LB_{1-5-3-6-4}^{4} = 23 + 0 + 1 = 24$$

$$LB_{1-5-3-6-4}^{5} = \max\left\{23, \min\{9,22\}\right\} + \max\left\{1, \frac{1-\max\{0, \max(23,9,22)-\max(23,\min(9,22))\}}{2}\right\} = 24$$

$$LB_{1-5-3-6-4} = \max\{35,\ 29,\ 26,\ 24,\ 24\} = 35$$

$Global\_LB\ [LEVEL(5)] = \min\{30,\ 35\} = 30$,

branch from node 19

*LEVEL 6*

The lower bounds on node 1-5-3-6-2-4 as follows:

$$LB_{1-5-3-6-2-4}^{1} = 29+0+0 = 29$$

$$LB_{1-5-3-6-2}^{2} = 29+0+0 = 29$$

$$LB_{1-5-3-6-2-4}^{3} = 29+0 = 29$$

$$LB_{1-5-3-6-2-4}^{4} = 29+0+1 = 30$$

$$LB_{1-5-3-6-2-4}^{5} = \max\left\{29, \min\{9,22\}\right\} + \max\left\{1, \frac{1-\max\{0, \max(29,9,22)-\max(29,\min(9,22))\}}{2}\right\} = 30$$

$$LB_{1-5-3-6-2-4} = \max\{29,\ 29,\ 29,\ 30,\ 30\} = 30$$

$C_{\max} = 30$

Branching scheme of the numerical example is given in Figure B.1.

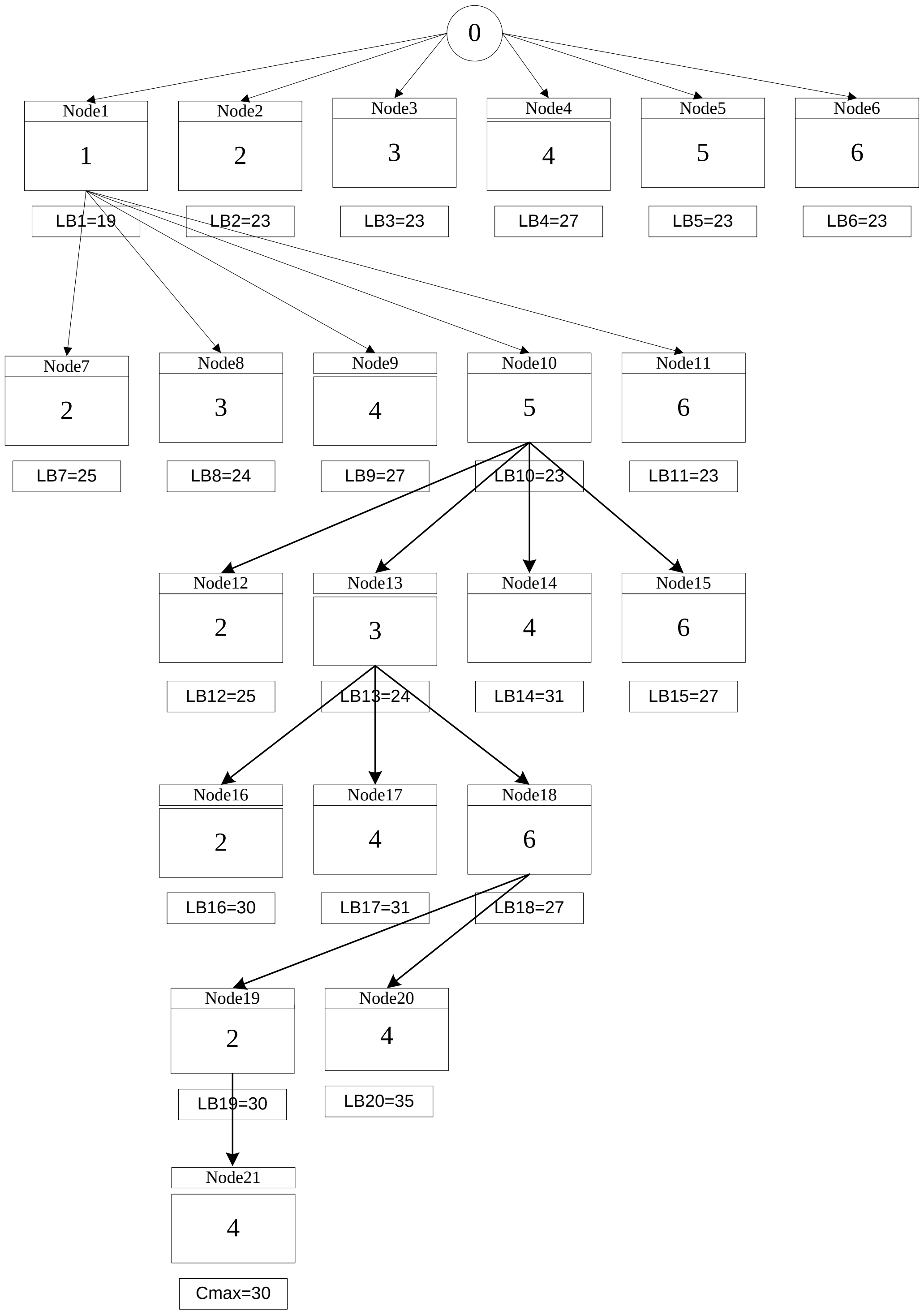

**Figure B.1: Branching Scheme of the Numerical Example**