DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A MODIFIED SHIFTING BOTTLENECK HEURISTIC FOR THE REENTRANT JOB SHOP SCHEDULING PROBLEM

by
Gamze KILINÇLI

April, 2007
İZMİR

A MODIFIED SHIFTING BOTTLENECK HEURISTIC FOR THE REENTRANT JOB SHOP SCHEDULING PROBLEM

A Thesis Submitted to the
Graduate School of Natural Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science in Industrial Engineering, Industrial Engineering Program

by
Gamze KILINÇLI

April, 2007
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A MODIFIED SHIFTING BOTTLENECK HEURISTIC FOR THE REENTRANT JOB SHOP SCHEDULING PROBLEM"** completed by **GAMZE KILINÇLI** under supervision of **ASST. PROF. DR. ŞEYDA TOPALOĞLU** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Şeyda TOPALOĞLU

Supervisor

Öğr. Gör. Dr. Gökalp YILDIZ

(Jury Member)

Doç. Dr. Ender Yazgan BULGUN

(Jury Member)

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and gratitude to Asst. Prof. Dr. Şeyda TOPALOĞLU for her academic guidance and enthusiastic encouragement throughout the research. I would also like to express my special thanks again to her for giving me the opportunity to share her experience and insights during this process that inspired creativity and motivation.

I am grateful to all of the personnel of the textile company who helped me to gather the data of the study.

I wish to thank my parents and my sister who have been always an important source of support, encouragement and endless love. I also would like to mention the priceless contributions of H. Tanju TAŞKIRAN.

Gamze KILINÇLI

A MODIFIED SHIFTING BOTTLENECK HEURISTIC FOR THE REENTRANT JOB SHOP SHEDULING PROBLEM

ABSTRACT

In recent years there has been a great increase of interest in the job shop scheduling problem (JSSP) from both practical and theoretical points of view. This is due to the large amount of computational effort usually required in the solution of the problem. Several heuristics that lead to near-optimal solutions in reasonably short times have been developed to solve this problem. Among the heuristics, one of the most successful approaches is the shifting bottleneck heuristic (SBH) which is a disjunctive graph-based decomposition method. Recently, the reentrant job shop has emerged as a new type of manufacturing shop. This problem is important because the assumption that each job visits each machine only once is often violated in practice. The principle characteristic of a reentrant job shop is that a certain job may visit a specific machine or a set of machines more than once during the process flow. In this study, a modified shifting bottleneck heuristic (MSBH) is proposed to solve the reentrant job shop scheduling problem (RJSSP) for makespan minimization objective. Furthermore, a modified rolling horizon procedure (MRHP) is proposed to decompose large job shops. Finally, various comparative experiments and an application study in a textile factory have been carried out to evaluate the performance of both the MSBH and the MRHP.

Keywords: Reentrant Job Shop Scheduling Problem, Shifting Bottleneck Heuristic, Rolling Horizon Procedure, Makespan Minimization.

YENİDEN GİRİŞLİ İŞ ATÖLYESİ ÇİZELGELEME PROBLEMİ İÇİN ÖNERİLEN DARBOĞAZ KAYDIRMA SEZGİSEL YÖNTEMİ

ÖZ

Son yıllarda iş atölyesi çizelgeleme problemine hem pratik hem de teorik açıdan gösterilen ilgi artmıştır. Bunun nedeni problemin çözümü için gerekli hesaplama çabasının oldukça fazla olmasıdır. Bu problemi çözmek için daha makul sürelerde optimuma yakın sonuçlar veren çeşitli sezgisel yöntemler geliştirilmiştir. Bu sezgisel yöntemler arasında en başarılı olanlarından biri şebeke diyagramı tabanlı bir ayrıştırma metodu olan darboğaz kaydırma sezgisel yöntemidir. Son dönemde, yeniden girişli iş atölyesi yeni bir imalat atölyesi çeşidi olarak ortaya çıkmıştır. Her işin her makinede sadece bir defa işlem gördüğü varsayımına pratikte genellikle uyulmadığı için bu problem önem kazanmaktadır. Yeniden girişli iş atölyesinin temel özelliği bir işin üretim süreci boyunca herhangi bir makinede birden fazla işlem görebilmesidir. Bu çalışmada, orijinal darboğaz kaydırma sezgisel yöntemi yeniden girişli iş atölyesi çizelgeleme problemi için revize edilmiştir. Amaç fonksiyonu en son işin tamamlanma zamanının enküçüklenmesidir. Ayrıca, büyük ölçekli iş atölyelerinin daha küçük bölümlere parçalanarak çözülebilmesi için bir sarmalı çevren prosedürü önerilmiştir. Son olarak da önerilen yöntemlerin performansları çeşitli iş - makine sayısı kombinasyonları için test problemleri ve bir tekstil fabrikasında uygulama çalışması kullanılarak karşılaştırılmıştır.

Anahtar sözcükler: Yeniden Girişli İş Atölyesi Çizelgeleme Problemi, Darboğaz Kaydırma Sezgisel Yöntemi, Sarmalı Çevren Prosedürü, En Son İşin Tamamlanma Zamanının Enküçüklenmesi.

CONTENTS

Page

THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT.....iv
ÖZ v

CHAPTER ONE – INTRODUCTION 1

CHAPTER TWO – THE JOB SHOP SCHEDULING PROBLEM 4

2.1 The Job Shop Scheduling Problem ……...…..4
2.1.1 The Mathematical Formulation And Graphical Representation ……..…6

2.2 The Reentrant Job Shop Scheduling Problem ……..……..…8
2.2.1 Disjunctive Graph For The Reentrant Job Shop Scheduling Problem 10

2.3 Methods Employed In Job Shop Scheduling Problems…….……..11
2.3.1 Exact Methods ..…….…….12
2.3.1.1 Integer Programming ……..……..…..…..12
2.3.1.2 Branch And Bound Algorithm ……..……...13
2.3.2 Approximation Algorithms …….....…....14
2.3.2.1 Priority Dispatching Rules ……..….15
2.3.2.2. The Shifting Bottleneck Heuristic …….…....16
2.3.2.3 Local Search ……..…..21
2.3.2.4 Computer Simulation …….…..…23

CHAPTER THREE – OVERWIEW OF THE SOLUTION APPROACHES...29

3.1 The Mixed Integer Programming Model For The Reentrant Job Shop Scheduling Problem ……..……...……....….29

3.2 The Shifting Bottleneck Heuristic ……....….....…33

3.3 The Rolling Horizon Procedure ……....….40

3.3.1 Assignment Of Operations To Time Windows ……..….41

CHAPTER FOUR – THE MODIFIED SHIFTING BOTTLENECK HEURISTIC…….....45

4.1 The Modified Shifting Bottleneck Heuristic For The Reentrant Job Shop Scheduling Problem ……...…….…….….45

4.1.1 The Modified Algorithm For Solving The Single Machine Maximum Lateness Problem ……...…….….50

4.1.2 The Modified Resequencing Procedure …….....52

4.1.3 The Improved Cycle Elimination Procedure ……..….53

4.1.4 The Improved Method For The Reentrant Flow …….54

4.2 The Modified Shifting Bottleneck Heuristic With Total Machine Load And Total Infeasibility Subproblem Criticality Measures ……....60

4.2.1 Total Machine Load …….....60

4.2.2 Total Infeasibility ……...…....61

CHAPTER FIVE – THE MODIFIED ROLLING HORIZON PROCEDURE.64

5.1 The Modified Rolling Horizon Procedure ……..……64

5.1.1 Assignment Of Operations To Time Windows …….....….....71

5.1.2 Optimization Of Each Time Window ……..…….….72

5.2 Improved Decomposition Heuristic For A Special Job Shop ……...….75

CHAPTER SIX – COMPUTATIONAL STUDY 77

6.1 Experimental Framework …….……..….77

6.2 Comparison Of The Modified Shifting Bottleneck Heuristic With The Original Shifting Bottleneck Heuristic For The Classical Job Shop Scheduling Problem.....79

6.2.1 Computational Results For Benchmark Problems ..……..…..……79

6.3 Comparison Of The Proposed Methods For The Reentrant Job Shop Scheduling Problem For Small Size Test Problems ……..……...…83

6.3.1 Details Of The Implementation And The Computational Results …..…85

6.4 Comparison Of The Proposed Methods For The Reentrant Job Shop Scheduling Problem For Large Test Problems ……..…..…….…….....118

6.4.1 Details Of The Implementation And The Computational Results …....119

6.5 Comparison Of The Proposed Methods For The Special Reentrant Job Shop Scheduling Problem ……...……..…….....….136

6.5.1 Details Of The Implementation And The Computational Results.....…137

6.6 Conclusion ……...……..……...….146

CHAPTER SEVEN – APPLICATION PROBLEM IN A TEXTILE FACTORY……..……148

7.1 Company Profile …….…....…….148

7.1.1 The Reentrant Job Shop Scheduling Problem In The Dyeing-Finishing Facility …….……..…….149

7.2 Solution Of The Application Problem ……..…….…..…….152

CHAPTER EIGHT – CONCLUSIONS 171

REFERENCES ……...……....173

APPENDIX A: The MATLAB Code Of The Modified Shifting Bottleneck Heuristic…….……...…...…192

APPENDIX B: The Computational Results Of The Test Problems ……..…196

APPENDIX C: Data For The Application Problem …….……..…...….221

CHAPTER ONE
INTRODUCTION

Timely and cost effective production is becoming increasingly important in today`s global competitive markets. Manufacturing and service organizations frequently face the challenge of making high quality products quickly and delivering those products to their customers on time. Improvements in the scheduling of their operations can often contribute to their success in meeting these goals. However, as manufacturing processes become more complex, the difficulty of finding good production schedules increases. Effective production scheduling techniques are particularly important in large, complex manufacturing systems.

Scheduling is the problem of allocating limited resources to operations over time. Therefore, it is a complex task that can be formulated using a constraint-based representation. Some instances of scheduling are course timetabling at school, examining patients at clinics, scheduling for maintaining and repairing production machines, and sequencing jobs that arrive at one or more work centers.

The classical job shop scheduling problem (JSSP) is regarded as one of the most well-known scheduling problems. As recognized in the literature, it is NP-hard and therefore computationally intractable.

The classical JSSP consists of scheduling a set of jobs on a set of machines under the constraints of limited resources, time and precedence requirements among operations of the jobs. The aim is to identify the job sequences on machines that optimize certain performance criteria subject to the constraints. Minimizing some function of the completion times of the jobs such as minimizing makespan, minimizing maximum lateness/tardiness, minimizing mean flow time are widely used in literature.

The problem of minimizing makespan in a job shop without reentrant product flow can be represented by a disjunctive graph. The disjunctive graph representation

forms the core of several heuristic procedures for the JSSP, such as the shifting bottleneck heuristic (SBH). The SBH is based on the idea of sequencing the machines one at a time. This approach iteratively identifies a bottleneck machine and sequences it optimally while holding the job sequence fixed on the machines already processed and ignoring the remaining machines. This is done by solving a single machine problem with release times and due dates. Once a machine has been sequenced, each machine in turn is freed up and resequenced cyclically. Furthermore, in the literature, the rolling horizon procedure (RHP) which results in good solutions for large job shops in short computational efforts are used to decompose the problems.

Recently, the reentrant job shop has emerged as a new type of manufacturing shop. This problem is important because the assumption that each job visits each machine only once is often violated in practice. The principle characteristic of a reentrant job shop is that a certain job may visit a specific machine or a set of machines more than once during the process flow.

The subject of this thesis is the reentrant job shop scheduling problem (RJSSP). The application problem is derived from a textile factory which operates on a reentrant flow basis. For the solution of this problem, the SBH and the RHP have been modified.

In Chapter 2, we describe both the classical and reentrant job shop scheduling problems. Mathematical formulations, objectives, solution procedures and application areas of these problems are articulated with various examples derived from the literature.

In Chapter 3, the mixed integer programming (MIP) formulation for the RJSSP is given initially. Following, as a solution approach for the classical JSSP, the SBH and the RHP are explained respectively, which will be the basis for our proposed solution approaches.

In Chapter 4, the original SBH is modified to solve the RJSSP for makespan minimization objective. This modified heuristic differs from the original SBH in some aspects such as the subproblem solution procedure and the resequencing step. Furthermore, Total Machine Load (TML) and Total Infeasibility (TOTINF) subproblem prioritization procedures are adapted to the modified shifting bottleneck heuristic (MSBH) to remove the need of solving all single machine problems at each iteration.

In Chapter 5, a modified rolling horizon procedure (MRHP) is developed for large reentrant job shops, in which the makespan must be minimized. The method divides the problem into a number of subproblems each corresponding to a time window of the overall schedule. These subproblems are solved using the MSBH with Total Machine Load (MSBH_TML). Furthermore, a number of rules for assignment of operations to time windows are derived. In addition, we develop a modified decomposition procedure for the special job shop which is the application problem of this study in the associated textile plant.

In Chapter 6, the proposed methods are tested on various randomly generated test problems and benchmark problems from the literature. The aim of this computational study is to see the effectiveness of the proposed methods in comparison to the original methods, and to compare the methods among themselves. Furthermore, to see the effectiveness of these models, the solution qualities and times are compared for different rescheduling levels and for different number of time windows.

In Chapter 7, the RJSSP of the textile factory is described and solved by the proposed methods. The factory produces a large variety of manufactured fabric, and works under a make-to-order policy in a complex process ranging from yarn producing to yarn dyeing, weaving and dyeing-finishing. We schedule the dyeing-finishing facility of this factory which is a special RJSSP having the first and last operations of all the jobs performed on the same machines, and the scheduling horizon is taken as one month.

CHAPTER TWO
THE JOB SHOP SCHEDULING PROBLEM

In this chapter, we describe both the classical and reentrant job shop scheduling problems. Mathematical formulations, objectives, solution procedures and application areas of these problems are articulated with various examples derived from the literature.

2.1 The Classical Job Shop Scheduling Problem

The classical job shop scheduling problem (JSSP) consists of scheduling a set of jobs on a set of machines under the constraints of limited resources, time and precedence requirements among operations of the jobs. The processing of a job on a machine is called an operation, and it has a constant duration. The aim is to identify the job sequences on machines that optimize certain performance criteria subject to the constraints. The constraints are as follows:

- The precedence relationships of the operations on each job must be respected,
- Each machine can process at most one job at a time,
- Preemption is not allowed, that is, once a machine starts processing an operation, it can not be interrupted.

The additional assumptions which are used in the classical JSSPs are:

- The processing times are independent of the sequence,
- All jobs are ready for processing at time zero at which the machines are idle and immediately available for work,
- Machines are available throughout the scheduling period,
- There is only one of each type of machine,
- In-process inventory is allowed,
- There is no randomness; all the data are known and fixed.

In general, one of the following objectives may exist in the JSSPs:

- To maximize the utilization of systems or resources, in other words to minimize the floating time of jobs,
- To minimize the operational or idle time in each machine, caused by earliness of operation process or tardiness in receiving the jobs by a machine.

Minimizing some function of the completion times of the jobs such as minimizing makespan, minimizing maximum lateness/tardiness, minimizing mean flow time are widely used in literature. Most research on job shop scheduling to date has focused on the problem of minimizing makespan (C_{max}), denoted by $J \mid\mid C_{max}$ in the notation of Graham et al. (1979). Some other factors that affect the JSSP are setup time, release date, due date, batching, transportation, etc.

The JSSP is a class of combinational optimization problems known as NP-hard one (Lenstra et al., 1977). In a JSSP with n jobs and m machines, there is $(n1)!(n2)!...(nm)!$ sequence alternatives. Definitely, all these alternatives may not feasible. However, it is nevertheless impossible to evaluate all feasible sequences within a reasonable amount of time. Since each job type follows different processing routes, the JSSP constitutes one of the hardest scheduling problems and it is often very difficult to find an optimal solution. Despite this difficulty, the JSSP has been a major target for many researchers in the last three decades due to its wide practice in real world.

The JSSPs are formulated as integer programs. In the next section, the problem of minimizing the makespan in a job shop without reentrant product flow is formulated by an integer programming formulation and represented by a disjunctive graph.

2.1.1 The Mathematical Formulation And Graphical Representation

A review of the literature reveals that the integer programming formulations of Wagner (1959), Bowman (1959) and Manne (1960) are widely used for the JSSPs. The integer programming formulation of Manne (1960)' s model is described below.

The notation used in the model is as follows:

n	number of available jobs for processing at time zero
m	number of machines in the shop
J_i	job number i
M_k	machine number k
p_{ik}	processing time of J_i on M_k
r_{ilk}	1 if the l th operation of J_i requires M_k; 0 otherwise
x_{ik}	starting time of J_i on M_k
C_i	completion time of J_i
C_{max}	maximum completion time or makespan; $C_{\max} = \max_{i=1}^{n} C_i$

The model is as follows:

Minimize $C_{\max}$ (1.1)

Subject to

$$\sum_{k=1}^{m} r_{ilk}\left(x_{ik} + p_{ik}\right) \le \sum_{k=1}^{m} r_{i,l+1,k}\, x_{ik} \qquad i = 1,2,...,n,\ l = 1,2,...,n-1 \quad (1.2)$$

$$\left(M + p_{jk}\right) y_{ijk} + \left(x_{ik} - x_{jk}\right) \ge p_{jk} \qquad 1 \le i < j \le n,\ k = 1,2,...,m \quad (1.3)$$

$$\left(M + p_{ik}\right)\left(1 - y_{ijk}\right) + \left(x_{jk} - x_{ik}\right) \ge p_{ik} \qquad 1 \le i < j \le n,\ k = 1,2,...,m \quad (1.4)$$

$$\sum_{k=1}^{m} r_{imk}\left(x_{ik} + p_{ik}\right) \le C_{\max} \qquad i = 1,2,...,n \qquad (1.5)$$

$$x_{ik} \ge 0\,,\ y_{ijk} = 0 \text{ or } 1 \qquad i, j = 1,2,...,n\,,\ k = 1,2,...,m \qquad (1.6)$$

Constraint set (1.2) ensures that the starting time of J_i on machine M_{k+1} should be later than its finish time on machine M_k. Constraint sets (1.3) and (1.4) satisfy the requirement that only one job may be processed on a machine at any instant in time. That is, either $x_{ik} - x_{jk} \ge p_{jk}$ or $x_{jk} - x_{ik} \ge p_{ik}$. Because y_{ijk} is a 0-1 integer and adding M, a large enough positive number, has the effect of eliminating the associated constraint, (1.3) and (1.4) together guarantee that one of the constraints must hold while the other is eliminated. Constraint set (1.5) gives the definition of C_{max} which is to be minimized in the objective function (1.1). The non-negativity and 0-1 restrictions for x_{ik} and y_{ijk}, respectively, are specified in (1.6).

The problem of minimizing the makespan in a job shop without reentrant product flow can be represented by a disjunctive graph. Consider a directed graph G with a set of nodes N and two sets of arcs A and B, which is denoted by $G=(N,A,B)$. The processing of job j on machine i is referred to as operation (i,j) and its duration is p_{ij}. The nodes N correspond to all the operations (i,j) that must be performed on the n jobs. The conjunctive (solid) arcs A represent the routes of the jobs. If arc $(i,j)\rightarrow(k,j)$ is part of A, then, operation (i,j) precedes operation (k,j). Two operations that belong to two different jobs and have to be processed on the same machine are connected to one another by two so-called disjunctive (broken) arcs going in opposite directions. The disjunctive arcs B form m cliques of double arcs, one clique for each machine (clique is a term in graph theory that refers to a graph in which any two nodes are connected to one another; in this case each connection within a clique is a pair of disjunctive arcs). All arcs emanating from a node have a length of the processing time of the operation that is presented by that node (Pinedo & Chao, 1999).

A feasible schedule corresponds to a selection of one disjunctive arc from each pair, which determines the sequence in which the operations are to be performed on

that machine. The problem of minimizing the makespan is reduced to finding a selection of disjunctive arcs that minimizes the length of the longest path (that is, the critical path). The disjunctive graph representation forms the core of several heuristic procedures for the JSSP, such as the shifting bottleneck heuristic (SBH) which is the focus of this study.

An example of a disjunctive graph representation is given in Figure 2.1. The machine sequence and the processing times are given in Table 2.1.

Table 2.1 The data for the illustrative example

Jobs	Machine Sequence	Processing Times
1	1,2,3	p_{11}=10, p_{21}=8, p_{31}=4
2	2,1,4,3	p_{22}=8, p_{12}=3, p_{42}=5, p_{32}=6
3	1,2,4	p_{13}=4, p_{23}=7, p_{43}=3

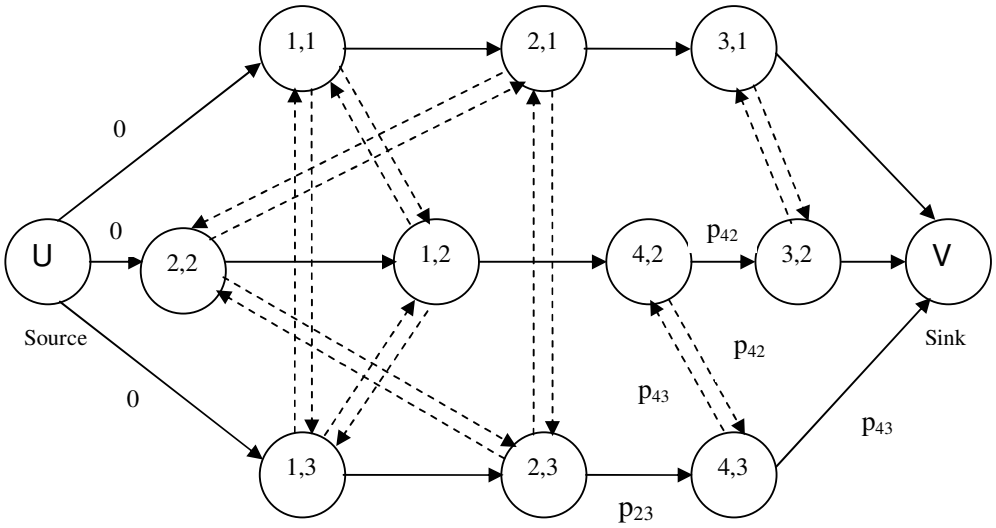


Figure 2.1 Directed graph for the job shop with the makespan as the objective

2.2 The Reentrant Job Shop Scheduling Problem

Recently, the reentrant job shop has emerged as a new type of manufacturing shop. This problem is important because the assumption that each job visits each machine only once is often violated in practice. The principle characteristic of a

reentrant job shop is that a certain job may visit a specific machine or a set of machines more than once during the process flow.

In general, the special assumptions of the RJSSPs are as follows:

- Each job may visit certain machines more than once,
- Any two succeeding operations of each job have to be processed on different machines,
- Each machine has different capabilities with respect to performing the same operation.

Reentrant job shops can be found in many production systems, particularly in high-tech industries such as semiconductor manufacturing. The capital-intensive nature of semiconductor manufacturing requires the existence of a reentrant processing environment. Therefore, rather than employing a machine entirely to a particular step, wafers usually make multiple visits to a machine for multiple step processing. In a reentrant line, wafers may spend a substantial amount of time waiting for an available machine which leads to long cycle times as well as low production volumes.

Another example might be the manufacturing of printed circuit boards that require both surface-mounted devices and conventional pin-through-hole devices. Also it is employed in parts that go through the painting and baking divisions alternately for different coats of paint in a painting shop. Likely, in the textile factory, which is the application job shop of this study, the fabrics are processed on the same machines several times through their process flows. For example, following some of the operations, the fabrics need to go through the same washing and drying machines on which they were processed before.

The makespan objective function is rarely studied in literature for the RJSSPs. Presumably the problem of minimizing makespan in a reentrant job shop is theoretically challenging.

2.2.1 Disjunctive Graph For The Reentrant Job Shop Scheduling Problem

The disjunctive graph representation for the classical JSSP with makespan minimization objective ($J_m \mid\mid C_{max}$) can be extended to the RJSSP with makespan minimization objective ($J_m \mid recrc \mid C_{max}$). If two operations of the same job have to be performed on the same machine, these two operations are not connected by a pair of disjunctive arcs as in the classical JSSP because they are already connected by conjunctive arcs because of the precedence relationships. The precedence constraints are the routing constraints on the different operations of the same job to be processed on the machine. Therefore, the set of disjunctive arcs of a machine will not form a clique if this machine is subject to reentrance.

Consider the illustrative example and Figure 2.1 which was given in the previous section, and suppose that the machine sequence of job 2 is 2, 1, 4, 3, 4 rather than 2, 1, 4, 3. Figure 2.2 shows the graph of the instance containing reentrant flow for job 2. Both the third and fifth operations of this job require processing on machine 4. While the node pairs {(4,2a), (4,3)} and {(4,2b), (4,3)} are connected by disjunctive arc pairs, the node pair {(4,2a), (4,2b)} is not. Each job's specified process routing requires that conjunctive arcs connect the nodes associated with the job. In Figure 2.2, operation (4,2a) precedes operation (4,2b) due to the job's process routing; therefore, the node pair {(4,2a), (4,2b)} is not connected with a disjunctive arc pair. The existence of reentrant flow through machine 2 precludes the formation of clique by nodes (4,2a), (4,2b) and (4,3).

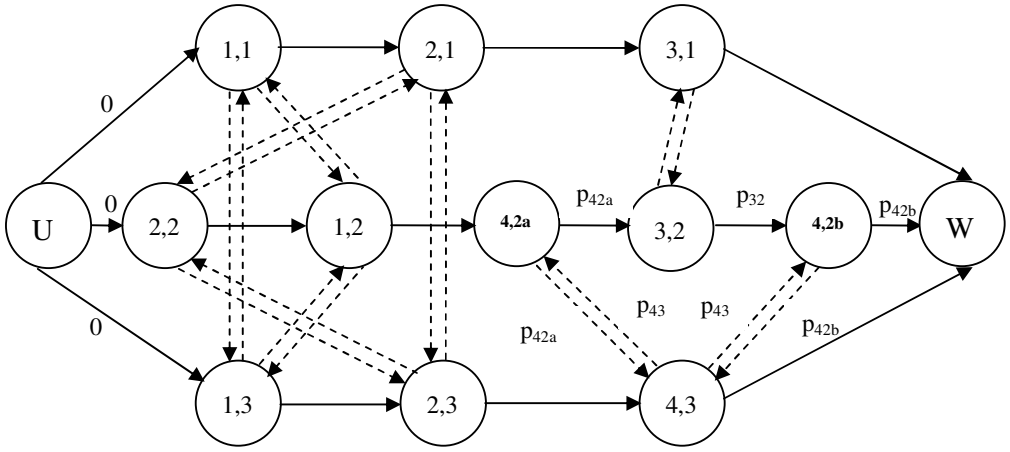


Figure 2.2 Disjunctive graph for a RJSSP

In very general terms, two types of techniques have been developed for the JSSPs; exact methods and approximation algorithms (heuristics). While mathematical programming models provide optimal solutions, they become too complex to model for large scheduling problems. On the other hand, heuristic approaches are simpler and very often independent of the problem size, but provide "good" rather than optimal solutions. In the next section, these methods and their applications in the literature are described in detail.

2.3 Methods Employed In Job Shop Scheduling Problems

A variety of scheduling rules for certain types of job shops have been considered in an enormous number of publications in the literature. The books by Muth & Thompson (1963), Conway et al. (1967), Baker (1974) and French (1982) as well as Rinnooy Kan (1976) can be referred. Dobson & Karmarkar (1986) presented formulations and decomposition approaches for a number of the JSSPs. Blazewicz et al. (1996) presented an overview of solution techniques for solving the JSSPs. Jain & Meeran (1999) reviewed the deterministic JSSP. In general, for a JSSP undermentioned approaches can be used to solve the problem. These approaches are outlined under two main techniques; exact methods and approximation algorithms.

2.3.1 Exact Methods

The exact methods for the JSSPs include integer programming formulations and branch and bound algorithms which are used to find optimal solutions. These methods have made considerable achievement, however, their implementation needs too much computational cost.

2.3.1.1 Integer Programming

With the advances of powerful computer capacity and efficient integer programming software, mathematical programming-based scheduling research is beginning to receive more and more attention from researchers. The use of integer programming in solving the JSSPs can be traced back to Wagner (1959).

Most integer programming problems arising in scheduling involve mixed integer programming, i.e., some variables are binary and others are continuous. The major contributions in mixed integer programming formulations for the classical JSSP are attributed to Wagner (1959), Bowman (1959), Manne (1960), and Wilson (1989). Manne (1960)' s formulation seems to involve considerably fewer variables than Wagner (1959) and Bowman (1959). Many scheduling textbooks (Baker, 1974; Conway et al., 1967; French, 1982) introduced integer programming formulations using Manne (1960)' s model for the JSSPs. Fisher et al. (1983) presented two mathematical formulations for this problem together with a solution strategy using a surrogate duality relaxation. The first formulation follows that of Manne (1960) and the second has been introduced in Fisher (1973).

Nemhauser & Wolsey (1988) presented a mathematical programming formulation for the JSSP with the objective of minimizing the sum of completion times. A survey on recent development of mathematical programming formulations for the JSSP can be found in a review article of Blazewicz et al. (1991) which ease the task of model building and testing scheduling formulations. Liao & You (1992) developed a new formulation that involves three major improvements on Manne (1960)' s model, each

representing a significant reduction of the computation time. These improvements are reducing functional constraints, imposing an upper bound on makespan and adding a lower bound constraint on makespan. Morten & Pentico (1993) presented discrete time formulations for the JSSPs by following Bowman (1959)' s binary variable definitions. Pan (1997) gave a study of integer programming formulations for the JSSPs. They compare Wagner (1959)' s, Bowman (1959)' s, Manne (1960)' s, Wilson (1989)' s and Morten & Pentico (1993)' s models based on the number of binary variables, constraints and continuous variables. They conclude that Manne (1960)' s model is the best among the five because it has the least number of binary variables.

Pan & Chen (2005) proposed formulations based on the models of Manne (1960) and Liao & You (1992) for the RJSSP. One of these formulations is used in this thesis to compare the proposed heuristics' results with the optimal values, which is described in detail in Chapter 3.

2.3.1.2 Branch And Bound Algorithm

One of the popular solution techniques for the JSSPs is the branch and bound method. The branch and bound approach is based on complete enumeration and is guaranteed to lead to an optimal schedule. The principal of this method is the enumeration of all feasible solutions of a combinatorial optimization problem such that properties or attributes not shared by any optimal solution are detected as early as possible.

To find exact solutions to the JSSPs, several branch and bound algorithms have been developed. Charlton & Death (1969), Baker & Su (1974), Rinnooy & Kan (1976), Lageweg et al. (1977) and Garey & Johnson (1979) proposed branch and bound methods to solve the JSSPs. Ashour et al. (1973) proposed a branch and bound algorithm based on the graph-theoretical representation of the JSSP for implicitly producing an optimal schedule such that the schedule time is minimized. For many years the algorithm by McMahon & Florian (1975) has been the most

efficient one. Others were less successful or led to improvements only in some special cases.

Balas (1969) and Carlier & Pinson (1989) presented branch and bound procedures based on the disjunctive graph representation of the JSSP. The first algorithm which solved the 10x10 benchmark problem of Muth & Thompson (1963), that remained unsolved for 25 years despite various attempts to solve it, and proved optimality of the solution was developed by Carlier & Pinson (1989).

Applegate & Cook (1991) developed lower bounds on the optimal makespan value using cutting planes and proposed a branch and bound algorithm. Brucker et al. (1994) and Martin & Shmoys (1996) used different problem formulations, branching schema and lower bound methods. Singer & Pinedo (1998) described a branch and bound method capable of solving 10 machines and 10 jobs instances. Al-Anzi et al. (2006) showed that the JSSP that minimizes total completion time can be modeled as finding the optimal coloring of a special mixed graph. They developed a branch and bound algorithm for obtaining the optimal coloring of such a mixed graph.

One of the main drawbacks of all branch and bound methods is the lack of strong lower bounds in order to cut branches of the enumeration tree as early as possible.

2.3.2 Approximation Algorithms

Because of the amount of computation involved, the exact algorithms can only be applied to problems of small sizes. Therefore, it is necessary to develop heuristics that lead to near-optimal solutions in a reasonably short time. Approximation algorithms use several approaches which can be classified as: priority dispatching rules, the SBH, local search strategies such as tabu search, simulated annealing and genetic algorithms, computer simulation, etc.

2.3.2.1 Priority Dispatching Rules

Priority dispatching rules are probably the most frequently applied heuristics for solving the JSSPs in practice due to their ease of implementation and their low time complexity. A dispatching rule is employed to select the next job to be processed among a group of jobs awaiting service such as First In First Out (FIFO), Last In First Out (LIFO), Shortest Processing Time (SPT), Most Work Remaining (MWKR), Earliest Due Date (EDD), etc.

The algorithm of Giffler & Thompson (1960) can be considered as a common basis of all priority rule based heuristics in job shop scheduling. An extensive discussion of dispatching rules can be found in Panwalkar & Iskander (1977) and Haupt (1989). Blackstone et al. (1982) and Bhaskaran & Pinedo (1991) reviewed recent studies of dispatching rules for the JSSPs.

Vepsalainen & Morton (1987) studied a number of priority rules such as EDD, Weighted Shortest Processing Times (WSPT), Cost Over Time (COVERT) and Apparent Tardiness Cost (ATC) for the weighted tardiness problem with job specific due dates and delay penalties. These penalties include customer badwill, cost of lost sales or changed orders and rush shipment cost. They conclude that the ATC rule achieves the best results. Vepsalainen & Morton (1988), considering the problem of minimizing total tardiness in a job shop, suggested a version of the ATC rule in which the slack of an operation is calculated using an estimate of the remaining flow time of the job in the shop.

Uzsoy et al. (1992) presented a computational study of dispatching rules for semiconductor testing operations for mean cycle time, mean tardiness, number of tardy jobs and maximum lateness (L_{max}) performance measures. The dispatching rules examined in the study are FIFO, SPT, Shortest Setup plus Processing Time (SSPT), Shortest Remaining Processing Time (SRPT), Earliest Job Due Date (Job-EDD), Earliest Operation Due Date (Operation-EDD), Cost Over Time (COVERT) and Apparent Tardiness Cost with Setups (ATCS). Their results indicate that no one

rule performs well for all performance measures. A follow-up study of them (Uzsoy et al., 1993) presents that for maximum lateness, the Operation-EDD rule has the best performance. Demirkol et al. (1995) developed extended dispatching procedures to develop improved schedules for complex job shops of the type encountered in semiconductor wafer fabs and testing facilities. Su et al. (1998) developed heuristic approaches based on various priority dispatching rules to solve the RJSSPs with maximum lateness and makespan objectives.

Holthaus & Rajendran (1997) presented five new and efficient dispatching heuristics as an additive combination of the process time, the total work-content of jobs in the queue of the next operation of a job, the arrival time and the slack of a job. By an extensive simulation study, they also observed that no single rule is effective in minimizing all measures of performance. Dabbas et al. (2001) proposed a modified dispatching approach that combines multiple dispatching criteria into a single rule with the objective of simultaneously optimizing multiple objectives. Jayamohan & Rajendran (2004) proposed dispatching rules by explicitly considering different weights or penalties for flow time and tardiness of a job.

2.3.2.2 The Shifting Bottleneck Heuristic

Among the heuristics, one of the most successful approaches is the SBH, a disjunctive graph-based decomposition method, proposed by Adams et al. (1988) to solve the JSSP with makespan minimization objective. The procedure sequences the machines one by one, successively, taking each time the machine identified as a bottleneck among the machines not yet sequenced. Every time after a new machine is sequenced, all previously established sequences are locally reoptimized. Both the bottleneck identification and the local reoptimization procedures are based on repeatedly solving certain single-machine scheduling problems. This heuristic is the basic method used in this thesis and will be described in Chapter 3 in detail.

The SBH of Adams et al. (1988) was extended to other performance measures like total weighted tardiness (Pinedo & Singer, 1996, 1999), maximum lateness

(Demirkol et al., 1997; Ovacik & Uzsoy, 1997) and number of tardy jobs (Yoo & MartinVega, 1997).

The original SBH was modified by both Dauzere-Peres & Lasserre (1993) and Balas et al. (1995), both of whom discuss the addition of delayed precedence constraints into the SBH to insure the feasibility of the single machine subproblem solutions. In the SBH, when the procedure fixes the sequence on a machine, it may thereby create a precedence constraint between some pairs of jobs on some unsequenced machines. It may be that an operation on a machine still to be scheduled must be processed after certain operations have been completed on that machine because of the sequences of the operations on the machines already scheduled. It may even be the case that, between the processing of two operations subject to these constraints, a certain minimum amount of time (that is, a delay) must elapse. These precedence constraints are therefore referred to as delayed precedence constraints.

The SBH decomposes the problem into subproblems which are solved according to some specified 'Subproblem Solution Procedure' (SSP), and then evaluated in terms of a specified performance or 'Machine Criticality Measure'. Applegate & Cook (1991) and Demirkol et al. (1997) showed that the quality of the overall solution obtained is affected by how the problem is decomposed, the order in which the subproblems are solved and the quality of the subproblem solutions. Holtsclaw & Uzsoy (1996) examined the effect of different indices for subproblem prioritization with the objective of minimizing the maximum lateness. Demirkol et al. (1997) found that better SSPs and reoptimization has higher solution quality. Aytug et al. (2002) proposed new measures of subproblem criticality and showed that they provide high quality solutions in less central processing unit time via computational experiments. Two of these new measures, Total Machine Load and Total Infeasibility, which are good measures for makespan problems, have been used in this thesis for the MSBH in Chapter 4.

Ivens & Lambrecht (1996) and Schutten (1998) examined several extensions of SBH towards real life applications with due dates, release dates, assembly structures,

split structures, overlapping operations, setup times, transportation times, parallel machines and beginning inventory. Jeong et al. (1999) modified the SBH to solve the JSSP with a before-arrival family setup time considering release date, transportation time and due date. Sun & Noble (1999) described applications of SBH to the JSSP with release dates, due dates and sequence-dependent setup times with the scheduling objective to minimize the weighted sum of squared tardiness.

Ramudhin & Marier (1996) generalized the SBH to solve various types of scheduling problems including job shops, open shops and assembly shops. Lawrence & Sewell (1997) examined the performance of SBH when applied dynamically to makespan problems while processing times are uncertain, and found that they give better results than optimal schedules applied in a static manner. Balas et al. (1998) proposed a job shop model with deadlines which must be met by all jobs and developed a heuristic procedure based on the original SBH. The objective is the makespan minimization. Most often in practice old jobs are to be completed within previously established deadlines and new jobs are to be simply completed as soon as possible. Their model depends on the need to add deadlines to the usual job shop framework in order to cope with this production model. Uzsoy & Wang (2000) examined the performance of several variants of SBH, some of which can be interpreted as naive implementations of the Theory of Constraints approach to scheduling. Their results show that SBH works better when well-defined bottleneck machines are present in the shop, and that problems become easier as the capacity difference between bottleneck and non-bottleneck machines increases. In Zhang et al. (2005), an efficient algorithm for the problem was presented with release and delivery times of each job, and based on this algorithm an improved SBH was developed. In Ohta & Nakatani (2006), a heuristic algorithm based on the SBH was proposed for solving the minimum total holding cost problem subject to no tardy jobs. Computational results indicate that the proposed algorithm performs well especially on the problem with tight due dates.

The SBH can also be used as a component of improving search heuristics, such as tabu search (Nowicki & Smutnicki, 1996; Pezzella & Merelli, 2000; Taillard, 1994),

simulated annealing (Matsuo et al., 1988; Van Laarhoven et al., 1992), local search (Balas & Vazacopoulos, 1998) and genetic algorithm (Dorndorf & Pesch, 1993, 1996). Wu et al. (2006) proposed a procedure to combine the fuzzy ranking method and the SBH to solve the JSSPs.

Many researchers have modified the SBH for the RJSSPs. Ovacik & Uzsoy (1992, 1997) extended the SBH to minimize maximum lateness with reentrant product flows, sequence dependent setup times and parallel machine workcenters for the scheduling of semiconductor manufacturing. They use a disjunctive graph representation to model the interactions between the subproblems. They report that the SBH applied to the test area of semiconductor wafer fabrication facilities leads to better results compared to pure dispatching approaches only if neighborhood search is included. Full reoptimization within the SBH is shown to consistently produce overall lower objective function schedules compared to when no reoptimization is performed. However, their work focuses on the specific industrial environment of semiconductor testing, making it difficult to generalize their results to other shop configurations.

Oey & Mason (2001) and Mason & Oey (2003) studied a complex job shop problem with reentrant flow and batch processing machines. A modified SBH is considered for generating machine schedules to minimize the total weighted tardiness. Since the modified SBH could produce infeasible schedules where cyclic schedules are found, a cycle elimination procedure is proposed to remove the possibility of the modified SBH generating cyclic schedules in the solution.

Wafer fabs were modeled as complex job shops by Mason et al. (2002), who extended the classical job shop work of Pinedo & Singer (1999) to develop a disjunctive graph formulation and modified SBH for minimizing the total weighted tardiness. They investigate the complex JSSP with reentrant product flow, batch processing possibilities, sequence dependent setups and parallel machine tool groups. Pfund et al. (2002) modeled semiconductor wafer fabrication process as a complex job shop and adapted the modified SBH of Mason et al. (2002) to facilitate the multi-

criteria optimization of makespan, cycle time and total weighted tardiness using a desirability function. The desirability function is implemented at two different levels of the modified SBH: the subproblem solution procedure level and the machine criticality measure level. A simulation framework that allows for the performance assessment of the modified SBH of Mason et al. (2002) in a dynamic environment via a rolling horizon approach was developed by Mönch et al. (2003). Pabst (2003) presented a modification of the SBH for scheduling wafer fabrication facilities.

In Mönch & DrieBel (2005), distributed versions of a modified SBH for complex job shops which contains parallel batching machines, machines with sequence-dependent setup times and reentrant process flows was considered. The used performance measure is total weighted tardiness. A two-layer hierarchical approach in order to decompose the overall scheduling problem is suggested. The upper (or top) layer works on an aggregated model. Based on appropriately aggregated routes it determines start dates and planned due dates for the jobs within each single work area, where a work area is defined as a set of parallel machine groups. The lower (or base) layer uses the start dates and planned due dates in order to apply SBH type solution approaches for the jobs in each single work area. Simulation experiments which are conducted in a dynamic job shop environment turn out that the suggested approach outperforms a pure First In First Out (FIFO) dispatching scheme and provides a similar solution quality as the modified SBH.

The proposed real options analysis model of Yeung & Mason (2006) which was developed to value the option to reoptimize in the SBH, effectively reduced the computation time required to perform the SBH in a complex job shop while still producing high-quality total weighted tardiness schedules. Mönch et al. (2007) considered a modified SBH for complex job shops which contains parallel batching machines, machines with sequence-dependent setup times and reentrant product flows. The performance measure is total weighted tardiness. They are interested in applying more sophisticated subproblem solution procedures like genetic algorithms for parallel machine scheduling in a dynamic job shop. Simulation experiments turn out that using near to optimal subproblem solution procedures leads in many

situations to improved results compared to dispatching-based subproblem solution procedures.

As to the efficiency of the SBH, a number of researchers have conducted computational experiments on the SBH where the performance measure to be minimized is makespan or maximum lateness (Adams et al., 1988; Applegate & Cook, 1991; Balas et al., 1995; Dauzere-Peres & Lasserre, 1993; Demirkol et al., 1997; Holtsclaw & Uzsoy, 1996; Lasserre, 1992; Ovacik & Uzsoy, 1992; Storer et al., 1992; Wenqi & Aihua, 2004). According to their conclusions, the SBH has higher solution quality in calculation times than that of other approaches and the performance of SBH is significantly affected by job routings. Furthermore, high quality subproblem solutions and reoptimization procedures are essential to obtaining good solutions. However, most of these efforts have focused on relatively small problems and small sets of benchmark problems available in the literature, making it difficult to relate specific problem characteristics to algorithm performance.

2.3.2.3 Local Search

Recently, local search strategies such as genetic algorithms, neural networks, tabu search, simulated annealing, constraint propagation, ejection chains, etc. are applied to the JSSP. A detailed overview of local search methods for the JSSP was given by French (1982), Morton & Pentico (1993), Vaessen et al. (1996) and Aarts & Lenstra (1997). Zhang & Huang (1995) presented a survey of applications of neural networks in the JSSPs.

Spachis & King (1979), Aarts et al. (1994) and Lourenco (1995) applied various local search techniques, while Storer et al. (1992, 1995), Croce et al. (1995) and Nose et al. (1999) used genetic algorithms based on problem space representations. Van Laarhoven et al (1992), Steinhöfel et al. (1999) and Satake et al. (1999) used simulated annealing approaches, while Dell' Amico & Turbian (1993), Taillard (1994), Barnes & Chambers (1995) and Nowicki & Smutnicki (1996) applied tabu search to this problem.

Sadeh (1991) used constraint-guided heuristic search to minimize the sum of tardiness and inventory costs. Blazewicz et al. (1996) concentrated on approximation algorithms belonging to a class of opportunistic and local search scheduling to solve practical size problems. Dorndorf & Pesch (1996) presented a number of genetic algorithms one of which is based on the SBH. Pezzella & Merelli (2000) proposed a local search method based on a tabu search technique and on the SBH used to generate the initial solution and to refine the next-current solutions. Watanabe et al. (2005) proposed a modified genetic algorithm with search area adaptation for solving the JSSP. Cavory et al. (2005) considered the cyclic JSSP and solved this problem by coupling of a genetic algorithm and a scheduler that utilizes a Petri-net modeling the linear precedence constraints between cyclic tasks. In Guo et al. (2006), a universal mathematical model of the JSSP for apparel assembly process was constructed and a genetic optimization process was then presented to solve this model, in which a new chromosome representation, a heuristic initialization process and modified crossover and mutation operators were proposed. The objective of this model is to minimize the total penalties of earliness and tardiness by deciding when to start each order's production and how to assign the operations to machines. Petrovic et al. (2006) presented a new tool which encompasses an interactive fuzzy multi-objective genetic algorithm which considers aspiration levels set by the decision maker for all the objectives for the multi-objective JSSPs. Ho et al. (2007) proposed architecture for learning and evolving of flexible job shop schedules called Learnable Genetic Architecture.

Several hybrid algorithms were developed using local search strategies. Lee et al. (1997) proposed to combine capabilities of genetic algorithms and machine learning techniques in order to develop a job shop scheduling system. Rahmati (1998) proposed a hybrid genetic algorithm for the non-classical JSSPs. Yu & Liang (2001) presented a hybrid approach combining genetic algorithms and neural networks to solve the expanded JSSP. Huang & Liao (2004) presented a hybrid algorithm combining ant colony optimization algorithm with the tabu search algorithm for the classical JSSP. To construct feasible schedules, the proposed algorithm employs a novel decomposition method inspired by the SBH, and a mechanism of occasional

reoptimizations of partial schedules. Gonçalves et al. (2005) presented a hybrid genetic algorithm for the JSSP. The chromosome representation of the problem is based on random keys. The schedules are constructed using a priority rule in which the priorities are defined by the genetic algorithm and improved by a local search heuristic. In Zhang et al. (2005), a heuristics search approach combining simulated annealing and tabu search strategy was developed for the JSSPs to overcome the problem that the quality of solutions found by tabu search approach depends on the initial solution.

In the literature, researchers used local search techniques to solve the RJSSPs. Cavalieri et al. (1999) presented an application of genetic algorithms to solve the RJSSP in a semiconductor manufacturing system that guarantees better performance than FIFO rule. Lam et al. (1999) and Nose et al. (1999) tried to solve the RJSSP using genetic algorithms.

2.3.2.4 Computer Simulation

By and large in the literature small size problems have been taken into account in application of the JSSPs. A few papers have used simulation methods for solving different problems. Shafaei & Brunn (1999) reported on a simulation study to investigate the performance of a number of scheduling rules on the basis of a rolling time horizon approach for a dynamic job shop environment. The performance measure considered is an economic objective which includes the main costs involved in a scheduling decision. Aydin & Öztemel (2000) proposed an intelligent agent based dynamic job shop scheduling system. It consists of two independent components: the agent and the simulated environment. Emre (2001) employed response-surface mapping methodology via regression analysis for reliable and static due date setting in multi-job shops. He uses the results obtained by computer simulation for the validation of setting the due date. In Tavakkoli-Moghaddam & Daneshmand-Mehr (2005), a simulation model was presented to work out the JSSPs with the objective of minimizing makespan. The model is coded by Visual SLAM which is a special simulation language based on the network modeling.

Computational results show that the simulation procedure is an efficient tool for solving the large scale JSSPs.

In addition to abovementioned studies in the literature, several algorithms have been improved to solve the JSSP and the RJSSP as given below.

In the literature, many authors were studied on disjunctive graph representation of the JSSP. Ashour & Parker (1971) concerned with the development of a precedence graph algorithm for solving the JSSPs. White & Rogers (1990) presented numerous extensions to the disjunctive graph formulation of the classic JSSP with makespan objective including assembly/disassembly operations, scheduled maintenance, sequence dependent setup times, and material handling delays. Blazewicz et al. (2000) proposed a new time and memory efficient representation of the disjunctive graph. It has a form of a graph matrix that combines advantages of a few classical graph representations enabling easy operation on the problem data.

Lagrangian relaxation is a mathematical programming technique for performing constrained optimization. In the literature, it is used to decompose a problem into subproblems. Hoitomt et al. (1993), Luh & Hoitomt (1993) and Baptiste et al. (in pres) used Lagrangian Relaxation to obtain approximate solutions to large job shop problems. Kaskavelis & Caramanis (1994, 1998) used efficient Lagrangian relaxation algorithms for the industry size job shop problems with more than 10,000 resource constraints in semiconductor testing facilities which have reentrant product flows.

Fuzzy set theory can be useful in modeling and solving the JSSPs with uncertain processing times, constraints, and setup times. Kuroda & Wang (1996) applied fuzzy theory to the real life JSSPs using fuzzy information regarding due dates and/or operation times, technological constraints, etc. Roobeek (1997) described a fuzzy logic based approach for increasing wafer fab throughput, called A Better Choice, in a job shop with reentrant process flows.

Anderson & Nyirenda (1990) and Raman & Talbot (1993) were interested in the job shop tardiness problem. Akpan (1996) introduced a network technique that offers a basis for finding an optimum solution for analysis of the JSSPs. In this technique, the jobs` priority on machines is determined by First Come First Serve (FCFS) rule. Lambrecht et al. (1998) proposed a four phase methodology to analyze and to schedule a job shop; a lot sizing and lead time estimation phase, a tuning phase, a scheduling phase and an execution phase. The objective is to minimize the expected lead time. The scheduling phase is based on an extended SBH.

Guinet (2000) proposed a procedure for the JSSPs with the objective of minimize the maximum completion time of jobs. Firstly, the JSSP is reduced to a flow shop problem by introducing job precedence constraints. Then, Johnson (1954)`s rule is applied to solve the problem. Singer (2001) presented a rolling horizon heuristic for large job shops, in which the total weighted tardiness must be minimized. The method divides a given instance into a number of subproblems, each having to correspond to a time window of the overall schedule, which is solved using SBH. Kaschel et al. (2002) introduced the result of an empirical study concerning the suitability of several methods such as priority rules, genetic algorithm, simulated annealing for the solutions of the JSSP. Lengyel et al. (2003) proposed a dynamic scheduling algorithm based on a feasibility function to schedule jobs in multi-machine random job shop. The purpose is to minimize unit penalty by achieving a balance between the number of tardy and early jobs, and reducing the difference between the maximum and the minimum lateness of jobs.

Alvarez-Valdes et al. (2004) described the design and implementation of a scheduling system in a glass factory which is a flexible JSSP and developed a heuristic algorithm to produce good quality solutions in short computing times. Chung et al. (2005) presented a heuristic algorithm that addresses a kind of JSSP with due date constraints by solving a series of smaller subproblems to optimality. The algorithm repeatedly executes in two steps, first improving the sequence of operations and second picking out the operations to be subcontracted on bottleneck machines. Akcora et al. (2005) proposed a heuristic for a JSSP of assigning

preemptive tasks with precedence constraints to flexible resources. The objective is to maximize the total reward in the job shop. Chandrasekaran et al. (2006) dealt with the criterion of the makespan minimization for the JSSP of different sizes. The proposed computational method of artificial immune system algorithm is used for finding optimal makespan values.

In the literature, there are various studies on the RJSSPs. Uzsoy et al. (1991) proposed algorithms for minimizing maximum lateness and minimizing number of tardy jobs in scheduling semiconductor test operations by decomposing the problem into a number of workstations and using a disjunctive graph representation. Kubiak et al. (1996) discussed the use of shortest processing time job order and dynamic programming algorithm to minimize the total flow time in a reentrant job shop. The RJSSP that they investigate has n jobs and k machines where each job enters the first machine (the hub) k times. Between any two consecutive entries into the hub, the job is processed on other machines.

Demirkol & Uzsoy (1997) evaluated the performance of decomposition procedures for scheduling complex job shops with reentrant product flows with respect to several different scheduling criteria in semiconductor testing facilities. They find that schedules developed to minimize maximum lateness also perform well with respect to makespan and total tardiness. Kuroda et al. (1999) proposed an efficient methodology to reduce production lead times observed in manufacturing systems with reentrant product flows. The methodology is to break up traditional job shops and restructure new manufacturing systems called cellular-line systems.

Odrey et al. (2001) presented a modeling approach for a reentrant flow manufacturing line in wafer fabs based on the use of generalized Petri nets. Kumar & Kumar (2001) provided an introduction to the application of queuing network models to the design and analysis of reentrant flow lines in semiconductor wafer fabs. Hierarchical production control approaches for reentrant semiconductor manufacturing lines were discussed in Vargas-Villamil & Rivera (2001), and Vargas-Villamil et al. (2003). Hopp et al. (2002) developed an optimized queuing network

capacity planning tool for supporting the design of semiconductor fabrication facilities that make use of queuing network approximations and an optimization route incorporating the common features of semiconductor environment such as batch processes, reentrant flows, multiple product class, etc.

In Miragliotta & Perona (2005), a new approach to the scheduling of reentrant shops was presented. Its main innovative principle is an objective driven engine: jobs to be processed are selected on the basis of a balanced evaluation of how well they fulfill efficiency and effectiveness objectives. This new approach relies on a heuristic algorithm build upon a decentralized architecture, in which each production resource can act as an independent decider and selects the jobs to process according to dynamically changing criteria and to widely shared information.

Mason et al. (2005) presented a mixed integer program heuristic to minimize total weighted tardiness in a complex job shop which has batching machines, parallel machines, machines with sequence-dependent setups, and reentrant product flows. The performance of the proposed heuristic is compared with the modified SBH of Mason et al. (2002) in terms of solution quality and execution speed, and with three dispatching rules using random problem instances from the literature. While the proposed heuristic produces superior schedules for problem instances with a small number of jobs, although it requires more execution time to arrive at these 'better' solutions, the SBH consistently outperforms the proposed heuristic for larger problem instances.

In Zoghby et al. (2005), modeling the RJSSP with sequence dependent setup times in the context of metaheuristic search methods was considered. In the study, reentrancy and its relationship to the traditional model is illustrated, and the new features of the disjunctive graph are introduced. Gupta & Sivakumar (2006) presented a brief review on job shop scheduling techniques in semiconductor wafer fabs that have reentrant flows.

The central component of most of these studies is the computational experiments executed to evaluate the performance of the algorithms proposed from the points of view of solution quality and computational requirements. Most of these experiments have been carried out for the minimizing makespan, maximum lateness and total weighted tardiness problems and have used relatively small sets of benchmark problems from the literature.

The difficulties in a JSSP with reentrant product flows are two fold. The first point is that there are a large number of processes in spite of several process types. The second point is a complicated material flow. Modified versions of the SBH are widely used in the literature to solve various types of the JSSPs. But, there is not a computational study of solving the RJSSP with the SBH for the makespan objective. In this thesis, the MSBH for the RJSSP with makespan objective will be proposed and applied to several test problems as well as the application problem itself.

As a result, the application problem of this thesis is different from the classical JSSP because of the reentrant product flows. Furthermore, the objective of the problem is minimizing makespan which is rarely studied for the RJSSP in the literature. The application plant has approximately 20 machines and 1000 jobs with 15-20 operations per job for a month. As it is seen, it is a large job shop. The other specific characteristic of the application problem is that the first operation of the jobs are identical and is processed on the same machine, likewise the last operation of the job are identical and processed on the same machine (the first and the last machine are not alike). So the heuristics used in the literature will need to be modified for our problem, which is described in detail in Chapters 4 and 5.

In the following chapter, methods employed in the JSSPs and the RJSSPs are described. The chapter will focus on mixed integer programming, SBH, and rolling horizon procedure.

CHAPTER THREE
OVERVIEW OF THE SOLUTION APPROACHES

In this chapter, the mixed integer programming (MIP) formulation for the reentrant job shop scheduling problem (RJSSP) is given initially. Following, the shifting bottleneck heuristic (SBH) and the rolling horizon procedure (RHP) are explained respectively, which will be the basis for our proposed solution approaches.

The optimal MIP model is utilized in comparison with the developed solution methods for small size test problems in Chapter 6. The SBH is considered as a starting point in the solution approach that we have developed for the RJSSP with makespan minimization objective. The RHP which results in good solutions for large job shops in short computational efforts has been modified for the makespan minimization objective and explained in Chapter 5.

3.1 The Mixed Integer Programming Model For The Reentrant Job Shop Scheduling Problem

With the possibility that a job may visit certain machines more than once, the reentrant job shop is a natural extension of the classical job shop. Pan & Chen (2005) described the development of a MIP formulation for the RJSSP based on an earlier classical job shop model developed by Manne (1960). This formulation is used to compare the results of proposed solution methods with the optimal values in this thesis.

Manne (1960) utilized binary variables to express the ‘either-or’ relationship for the non-interference restrictions for individual machine. To incorporate the reentry characteristic, Manne’s assumption that each job has exactly m operations and one for each machine has to be relaxed.

The notation used in the model is as follows:

m number of machines in the shop

n number of jobs for processing at time zero

J_i job number i

M_k machine number k

N_i number of operations of J_i

O_{ij} operation number j of J_i

p_{ij} processing time of O_{ij}

r_{ijk} 1 if O_{ij} requires M_k; 0 otherwise

s_{ij} the starting time of O_{ij}

C_{max} maximum completion time or makespan; $C_{\max} = \max_{i=1}^{n}\{C_i\}$

$Z_{iji'j'}$ 1 if O_{ij} precedes $O_{i'j'}$ (not necessarily immediately); 0 otherwise

H a very large positive number.

The model is as follows:

Minimize $C_{\max}$ (2.1)

Subject to

$$\sum_{k=1}^{m} r_{ijk}\left(s_{ij} + p_{ij}\right) \le \sum_{k=1}^{m} r_{i,j+1,k}\, s_{i,j+1} \qquad i=1,2,\ldots,n,\ j=1,2,\ldots,N_i\text{-}1 \qquad (2.2)$$

$$H\left(2 - r_{ijk} - r_{i'j'k}\right) + H\left(1 - Z_{iji'j'}\right) + \left(s_{i'j'} - s_{ij}\right) \ge p_{ij}$$

$$1 \le i \le i' \le n,\quad j=1,2,\ldots,N_i,\quad j'=1,2,\ldots,N_{i'},\quad k=1,2,\ldots,m \qquad (2.3)$$

$$H\left(2 - r_{ijk} - r_{i'j'k}\right) + HZ_{iji'j'} + \left(s_{ij} - s_{i'j'}\right) \ge p_{i'j'}$$

$$1 \le i \le i' \le n,\quad j=1,2,\ldots,N_i,\quad j'=1,2,\ldots,N_{i'},\quad k=1,2,\ldots,m \qquad (2.4)$$

$$\sum_{k=1}^{m} r_{i,N_i,k}\left(s_{i,N_i} + p_{i,N_i}\right) \leq C_{\max} \qquad i=1,2,\ldots,n \tag{2.5}$$

$$C_{\max} \geq 0,\ s_{ij} \geq 0 \qquad i=1,2,\ldots,n,\ j=1,2,\ldots,N_i$$

$$Z_{iji'j'} = 0 \text{ or } 1, \qquad 1 \leq i \leq i' \leq n,\ j=1,2,\ldots,N_i,\ j'=1,2,\ldots,N_{i'} \tag{2.6}$$

Constraint set (2.2) ensures that the starting time of $O_{i,j+1}$ on M_k is no earlier than the finish time of O_{ij}. Constraint sets (2.3) and (2.4) impose the requirement that only one job may be processed on a machine at any time. If both O_{ij} and $O_{i'j'}$ are processed on M_k, i.e., $r_{ijk} = r_{i'j'k} = 1$, then either $s_{i'j'} - s_{ij} \geq p_{ij}$ or $s_{ij} - s_{i'j'} \geq p_{i'j'}$ holds. Since $Z_{iji'j'}$ is a binary integer variable and H is a very large positive number, equations (2.3) and (2.4) together guarantee that one of the constraints must hold while the other is redundant. Constraint set (2.5) gives the definition of C_{max} to be minimized in the objective function (2.1). The nonnegativity constraints for C_{max} and s_{ij}, and the binary restrictions for $Z_{iji'j'}$ are altogether specified in (2.6).

An illustrative example is given below to understand the model. The machine sequences and the processing times are given in the Table 3.1. The optimal makespan is 44.

Table 3.1 The route and the processing times for the model

Jobs	Machine sequence	Processing times
1	2,3,1,3,1,3	p_{11}=10, p_{12}=1, p_{13}=9, p_{14}=3, p_{15}=7, p_{16}=6
2	1,2,3,2,3,1	p_{21}=5, p_{22}=6, p_{23}=8, p_{24}=9, p_{25}=2, p_{26}=3
3	1,3,2,1,2,1	p_{31}=4, p_{32}=4, p_{33}=5, p_{34}=1, p_{35}=5, p_{36}=7
Remarks: N_1=N_2=N_3=6		

The objective consists of the single variable C_{max}. The first set of constraints (2.2) consists of fifteen constraints: five for each job. For example, one of these constraints is

$$s_{11} + 10 \leq s_{12}$$

for job 1 and operations (1,1) and (1,2) on machines 2 and 3 consecutively.

The second and third constraint sets (2.3 and 2.4) consist of 324 constraints per set. An example for each set is

$$H * (1-Z_{1321}) + (s_{21}-s_{13}) \geq 9$$
$$H * Z_{1321} + (s_{13}-s_{21}) \geq 5$$

for job 1 and 2 and operations (1,3) and (2,1) on machine 1.

The fourth set of constraints (2.5) contains three constraints, one for each job. An example is

$$s_{16} + 6 \leq C_{max}$$

for job 1.

The number of binary restrictions for $Z_{iji'j'}$ is 324 and one of these is

$$Z_{1111} = 0 \text{ or } 1.$$

In this thesis, we will apply this aforementioned model to solve the RJSSP. For small size test problems, optimal results derived by this model are compared with the results of modified methods. A program coded in LINGO has been used to generate the model, and it has been solved with the branch and bound procedure of LINGO 8.0 on a PC P4/1.7G with 512 DRAM.

3.2 The Shifting Bottleneck Heuristic

Even though mathematical models yield in optimal results for the JSSP, as the size of the problem gets larger, the computational time increases substantially. In fact, mathematical models are insufficient for very large problem sizes. Therefore, various approximation algorithms have been developed for the solution of JSSP. As described in Section 2.3, when the literature is reviewed, we see that, one of the most successful heuristic procedures developed for the minimization of the makespan in a job shop is the SBH that is developed by Adams et al. (1998).

In the following overview of the SBH, M denotes the set of all m machines, and a subset M_0 of machines denotes a selection of disjunctive arcs that have been fixed at the previous iteration.

An iteration results in a selection of a machine from M-M_0 and the sequence in which the operations on this machine have to be processed. An unscheduled machine that causes the severest disruption is determined to include next in M_0. The original graph is modified by deleting all disjunctive arcs of the machines yet to be scheduled and keeping only the relevant disjunctive arcs of the machines in set M_0. The modified graph G' has one or more critical paths that determine the corresponding makespan $C_{max}(M_0)$.

Suppose the operation (i,j), $i \in \{M\text{-}M_0\}$, has to be processed in a time window of which the release date and due date are determined by the critical (longest) paths in G'. The release date is equal to the longest path in G' from the source to node (i,j), and the due date is equal to $C_{max}(M_0)$ minus the longest path from node (i,j) to the sink plus p_{ij}. Consider each machine in M-M_0 as a separate nonpreemptive single-machine problem which is NP-hard with release dates and due dates and with the maximum lateness (L_{max}) to be minimized. The minimum L_{max} of the single-machine problem corresponding to machine i is denoted by $L_{max}(i)$ and is a measure of the criticality of machine i.

After solving all these single-machine problems, the machine with the largest maximum lateness ($L_{max}(k)$) is chosen as the most critical or the "bottleneck" and therefore the one to be included next in M_0. According to the optimal solution, this machine is scheduled and the related disjunctive arcs are inserted in graph *G'*, then the makespan of the current partial schedule increases by at least $L_{max}(k)$; that is,

$$C_{\max}(M_0 \cup k) \geq C_{\max}(M_0) + L_{\max}(k)$$

Before starting the next iteration and determining the next machine to be scheduled, an additional step that resequences all the machines in the original set M_0 is done within the current iteration to see whether the makespan can be reduced. That is, a machine, say, machine *l*, is taken out of set M_0, and a graph *G"* is constructed by modifying graph *G'* through the inclusion of the disjunctive arcs that specify the sequence of operations on machine *k* and the exclusion of the disjunctive arcs associated with machine *l*. Machine *l* is resequenced by solving the corresponding single machine maximum lateness problem with the release and due dates determined by the critical paths in graph *G"*. Resequencing each machine in the original set M_0 completes the iteration.

In the next iteration, the entire procedure is repeated, and another machine is added to the current set $M_0 \cup k$.

The SBH is summarized in the undermentioned algorithm. It is given in Pinedo & Chao (1999) as follows:

Step 1. (Initial conditions)

Set $M_0=\emptyset$.

Graph *G* is the graph with all the conjunctive arcs and no disjunctive arcs.

Set $C_{max}(M_0)$ equal to the longest path in graph *G*.

Step 2. (Analysis of machines still to be scheduled)

Do the following for each machine *i* in set $M-M_0$:

Formulate a single-machine problem with all operations subject to release dates and due dates.

Minimize the L_{max} in each one of these single-machine subproblems.

$L_{max}(i)$ denotes the minimum L_{max} in the subproblem corresponding to machine i.

Step 3. (Bottleneck selection and resequencing)

Let

$$L_{\max}(k) = \max_{i \in \{M - M_0\}} (L_{\max}(i))$$

Schedule machine k according to the sequence obtained for it in step 2.

Insert all the corresponding disjunctive arcs in graph G.

Insert machine k in M_0.

Step 4. (Resequencing of all machines scheduled earlier)

Do the following for each machine $i \in \{M_0 - k\}$:

Delete the corresponding disjunctive arcs from G;

Formulate a single machine subproblem for machine i with release dates and due dates of the operations determined by longest path calculations in G.

Find the sequence that minimizes $L_{max}(i)$ and insert the corresponding disjunctive arcs in graph G.

Step 5. (Stopping criterion)

If $M_0=M$ then STOP; otherwise, go to step 2.

In summary, the general flowchart of the SBH is given in Figure 3.1.

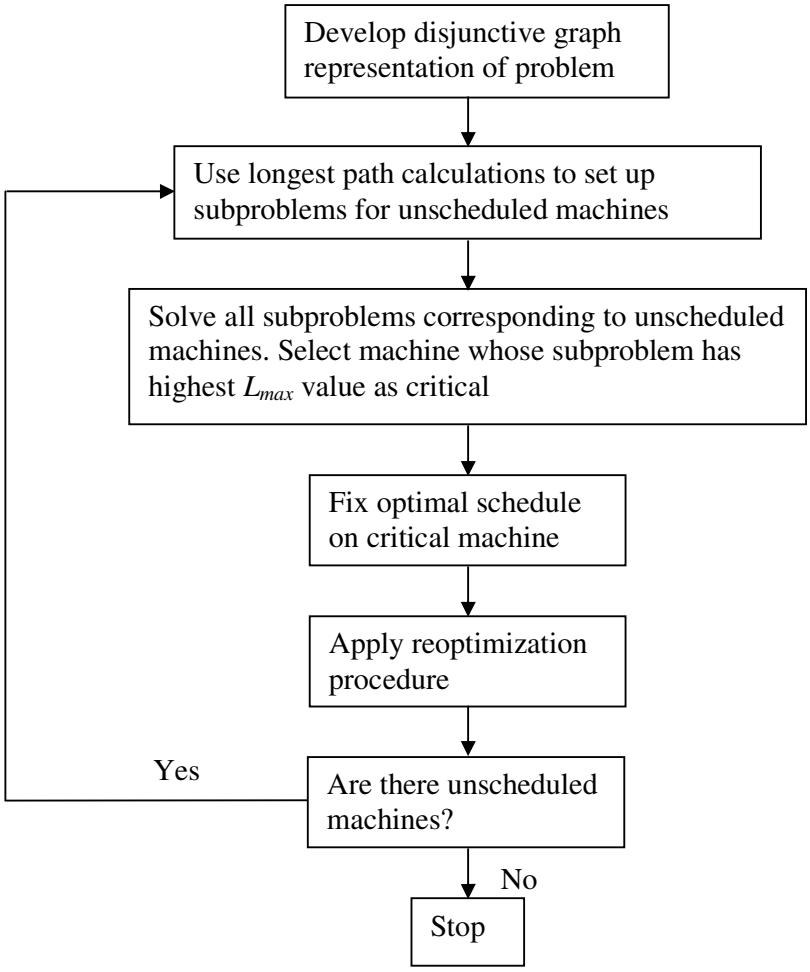


Figure 3.1 Flowchart of the Shifting Bottleneck Heuristic

In the remainder of this section, we present an example with four machines and three jobs that illustrate the use of the SBH. The machine sequences and the processing times are given in Table 3.2. p_{ij} denotes processing time, r_{ij} denotes release date and d_{ij} denotes due date of operation *(i,j)*.

Table 3.2 Machine sequences and processing times

Jobs	Machine Sequence	Processing Times
1	1,2,3	p_{11}=10, p_{21}=8, p_{31}=4
2	2,1,4,3	p_{22}=8, p_{12}=3, p_{42}=5, p_{32}=6
3	1,2,4	p_{13}=4, p_{23}=7, p_{43}=3

Iteration 1. Initially, set M_0 is empty, and *G'* contains only conjunctive arcs and no disjunctive arcs. The makespan $C_{max}(\emptyset)$ is equal to the maximum total processing time required for any job. By the critical path method, the maximum of 22 is achieved by both job 1 and job 2.

To determine which machine to schedule first, each machine is considered as a nonpreemptive single-machine maximum lateness problem with the release dates and due dates determined by the longest paths in *G'*.

The data for the nonpreemptive single-machine maximum lateness problem corresponding to machine 1 is determined as follows and the optimal sequence turns out to be 1, 2, 3 with $L_{max}(1) = 5$.

Jobs	1	2	3
p_{1j}	10	3	4
r_{1j}	0	8	0
d_{1j}	10	11	12

The data for the subproblem regarding machine 2 are given below and the optimal sequence for this problem is 2, 3, 1 with $L_{max}(2) = 5$.

Jobs	1	2	3
p_{2j}	8	8	7
r_{2j}	10	0	4
d_{2j}	18	8	19

Similarly, for machine 3 and 4, it can be shown that

$$L_{max}(3) = 4$$
$$L_{max}(4) = 0.$$

As a result, either machine 1 or machine 2 may be considered a bottleneck. Breaking the tie arbitrarily, machine 1 is selected to be included in M_0. The graph G'' is obtained by fixing the disjunctive arcs corresponding to the sequence of the jobs on machine 1 (see Figure 3.2). The makespan is 27.

$$C_{max}(\{1\}) = C_{max}(\emptyset) + L_{max}(1) = 22 + 5 = 27$$

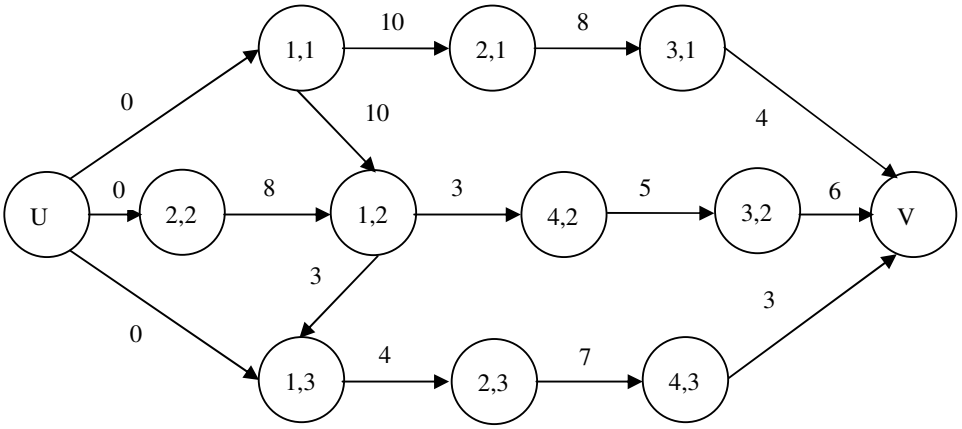


Figure 3.2 Iteration 1 of the Shifting Bottleneck Heuristic for the example

Iteration 2. Given that the makespan corresponding to G'' is 27, the critical paths in the graph can be determined. The three remaining machines have to be analyzed separately as nonpreemptive single-machine problems. The data for the problem concerning machine 2 are given below. The optimal schedule is 2, 1, 3 and the resulting $L_{max}(2) = 1$.

Jobs	1	2	3
p_{2j}	8	8	7
r_{2j}	10	0	17
d_{2j}	23	10	24

The data for the problem corresponding to machine 3 are given below. Both sequences are optimal and $L_{max}(3) = 1$.

Jobs	1	2
p_{3j}	4	6
r_{3j}	18	18
d_{3j}	27	27

Machine 4 can be analyzed in the same way, and the resulting $L_{max}(4) = 0$. Again, there is a tie, and machine 2 is selected to be included in M_0. So $M_0=\{1,2\}$, and

$$C_{max}(\{1,2\}) = C_{max}(\{1\}) + L_{max}(2) = 27 + 1 = 28.$$

The disjunctive arcs corresponding to the job sequence on machine 2 are added to *G''*, and graph *G'''* is obtained. As a part of iteration 2, an attempt may be made to decrease C_{max} ($\{1,2\}$) by resequencing machine 1. It can be checked that resequencing machine 1 does not yield any improvement.

Iteration 3. The critical path in *G""* can be determined, and machines 3 and 4 remain to be analyzed. These two problems both have a zero maximum lateness. Neither of the machines constitutes a bottleneck in any way.

The final schedule is determined by the following machine sequences: the job sequence 1, 2, 3 on machine 1; the job sequence 2, 1, 3 on machine 2; the job sequence 2, 1 on machine 3; and the job sequence 2, 3 on machine 4. The makespan is 28.

As described in this section, the SBH is very sensitive to the number of operations per machine. Although the results for small size problems are satisfactory, computational times grow exponentially for large size problems. So, for large JSSPs, RHPs have been developed by researchers. By limiting the number of jobs to be evaluated for scheduling on a machine, RHPs decompose large JSSPs into smaller, more tractable subproblems for which good solutions can be obtained. In the next section, this procedure and its application on an example is explained.

3.3 The Rolling Horizon Procedure

The RHP divides the problem into time windows that are optimized independently. As mentioned in Singer (2001), who presents a RHP for large job shops that minimizes the total weighted tardiness, RHPs becomes more effective with larger instances due to a scheme in which there is a double decomposition: the first one done horizontally in the Gantt chart using time windows, and the vertical one using the machine decomposition of the SBH. Furthermore, the different time windows are partially decoupled by using overlapping, while a disjunctive graph decouples the machines' subproblems within each window. The flow chart of the RHP is given in Figure 3.3.

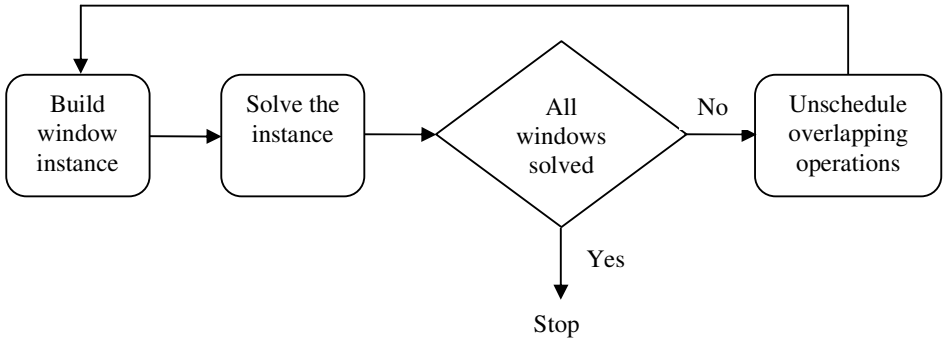


Figure 3.3 RHP in a job shop

In order to prevent the schedule for one time window from interfering with the next one, the time window has a degree of overlapping that is defined as follows. Let ω denote the size of the time window subproblem to be optimized, let $\phi \leq \omega$ denote the number of operations that will be "frozen" or scheduled according to the solution for the subproblem. Figure 3.4 shows an example where $\omega = 4$ and $\phi = 3$. In the first stage, a subproblem of four operations is optimized, yielding the sequence $A \rightarrow B \rightarrow C \rightarrow D$. In the second stage, the partial sequence $A \rightarrow B \rightarrow C$ is frozen and operations *D*, *E*, *F*, and *G* are optimized, and so on.

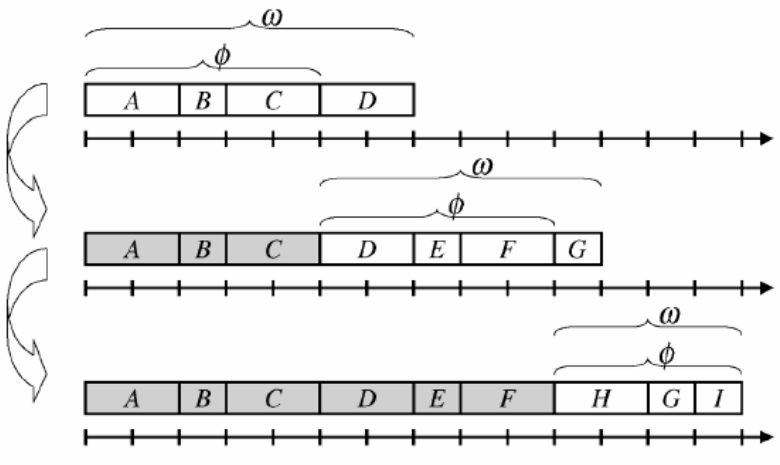


Figure 3.4 Overlapping in the rolling horizon procedure

The decomposition process will always make mistakes on assigning operations to each time window. Larger overlapping implies that the decomposition process, i.e., the assignment of operations to each time window, becomes less crucial since they overcome the mistakes of the decomposition process. However, a larger overlapping demands additional computing time since the windows become bigger.

In the RHP, the optimal number of windows to be used also grows with the size of the problem. Small instances present the best results with one or two windows, while larger instances are better solved with four and seven windows.

Several rules for the assignment of operations to time windows are given in Singer (2001) for the objective of minimizing total weighted tardiness. We will use modified versions of these rules in our proposed MRHP for makespan minimization objective.

3.3.1 Assignment Of Operations To Time Windows

In a RHP the subproblems correspond to time windows, where each window is strongly coupled with the previous one. In the case of two contiguous windows, the first one defines the machine availability for the operations in the second window, so it must be optimized first. The main issue is to define which operations belong to

each time window. Figure 3.5 shows a first decomposition schema that assigns the first operations of each job to the first time window for the illustrative example given in Table 3.3. Its explanation is that it divides a given instance into evenly sized smaller instances that keep their size under control, so that the time needed by the SBH does not grow too much.

Table 3.3 Input data for a total weighted tardiness job shop problem

Job	Machine and processing time			w_j	r_j	d_j
1	A 5	B 10	C 4	1	5	24
2	C 4	A 5	B 6	2	0	28
3	C 5	B 3	A 7	2	0	16

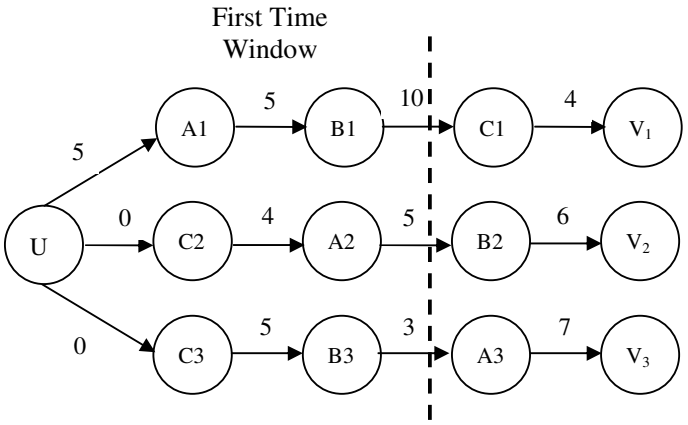


Figure 3.5 Time window with a fixed number of operations per job

A second scheme for assigning operations to the windows is to balance the number of operations per machine. Its justification is that the complexity of the SBH is very sensitive to the machine having the largest amount of operations since the subproblem complexity grows exponentially with the number of operations per machine.

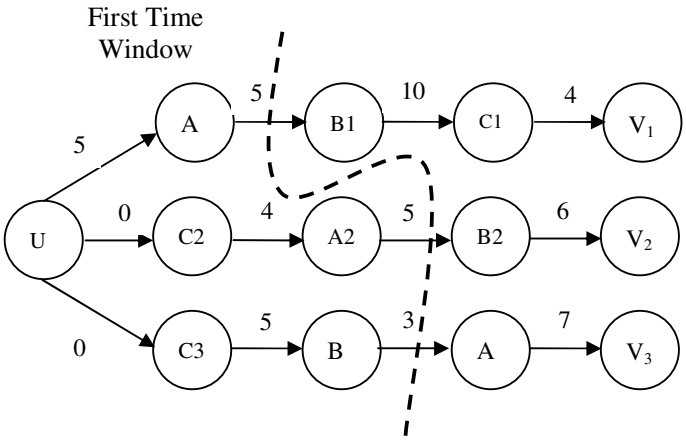


Figure 3.6 Time window with a balanced workload for each job

A third scheme, time window with a balanced workload for each job, is formalized as follows:

Define r_{jw} as the release time of the first operation of job j assigned to window w, and define the $workload_{jw}$ as the summation of the processing times of the operations of job j that are assigned to window w. Let ρ_{jw} denote the coefficient of variation of (r_{jw} + $workload_{jw}$) for all the jobs j that are assigned to window w. A time window is balanced if ρ_{jw} is below some given parameter, which is obtained through computational experiments.

Consider the example depicted by Figure 3.6 where the first window has a release time plus a workload of $r_1+p_{A1}=5+5=10$ for job 1, $r_2+p_{C2}+p_{A2}=0+4+5=9$ for job 2 and $r_3+p_{C3}+p_{B3}=0+5+3=8$ for job 3. The variance (σ^2) of the set {10,9,8} is 1 and its average (μ) is 9, so $\rho_{jw} = \sigma/\mu = 0.11$. If the window does not include operation $B3$ then the variance of the set {10,9,5} is 7 and its average is 8, so $\rho_{jw}=0.33$. So the first set is chosen for the first time window.

Singer (2001) concludes that, for the makespan objective function, it is likely that applying this technique would not be as efficient as that for the tardiness case. The total weighted tardiness function can be decomposed in an easier way than the

makespan since the former is associated with each operation, while the latter depends on a combination of all the operations in a simultaneous manner. So, we modified the RHP and the rules for assignment operations to time windows to meet the need of makespan objective. The modification effort and process is described in Chapter 5.

In this chapter, we explained the basic solution approaches for the JSSP. In the next chapter, the MSBH which is proposed for the RJSSP is described. The differences, advantages and disadvantages of the MSBH from the original SBH are clarified.

CHAPTER FOUR
THE MODIFIED SHIFTING BOTTLENECK HEURISTIC

In Chapter 2, the classical job shop scheduling problem (JSSP) and a variant of it, namely, the reentrant job shop scheduling problem (RJSSP) were described. As a solution approach for the classical JSSP, the original Shifting Bottleneck Heuristic (SBH) was explained in Chapter 3.

In this chapter, the original SBH is modified to solve the RJSSP for makespan minimization objective. This modified heuristic differs from the original SBH in some aspects such as the subproblem solution procedure and the resequencing step. Furthermore, Total Machine Load (TML) and Total Infeasibility (TOTINF) subproblem prioritization procedures are adapted to the modified Shifting Bottleneck Heuristic (MSBH) to remove the need of solving all single machine problems in each iteration.

4.1 The Modified Shifting Bottleneck Heuristic For The Reentrant Job Shop Scheduling Problem

The original SBH of Adams et al. (1988) decomposes the problem into subproblems involving a single machine and solves them one by one, successively. In each iteration, the machine with the largest maximum lateness (L_{max}) is scheduled as the bottleneck machine. The original SBH was developed for the classical JSSP. Upon literature review, it has been determined that the original SBH can be adapted and applied to more general job shop models like the ones that contain reentrant product flows, sequence dependent setup times and parallel machine workcenters.

In order to reduce the operation quantity as well as the computation time in the original SBH, a modified version of SBH has been developed for the RJSSP.

In the MSBH, as in the original SBH, the disjunctive graph of the problem is constituted as a first step. Conjunctive arcs are installed to the graph in order to

depict the precedence relationship among the operations and so are disjunctive arcs in order to depict the relationships among the operations to be processed on the same machine. Makespan is computed by Critical Path Method (CPM). The release dates and due dates for each operation in the subproblem are obtained using longest path calculations in the directed graph.

As far as solving the single machine problem is concerned, the original SBH and the MSBH fall apart. To solve the single machine L_{max} problem, while the original SBH uses the branch and bound type algorithm, we use a modified L_{max} algorithm which is described in the remainder of this section. In the next step, as in original SBH, the machine with the maximum L_{max} is chosen as the bottleneck. The sequence obtained for it is scheduled and all the corresponding disjunctive arcs are inserted in graph.

In the resequencing step, for each machine, which has been scheduled in former iterations, the corresponding disjunctive arcs are deleted from graph successively. A single machine subproblem for that machine is formulated and the sequence that minimizes L_{max} is found. Then, the corresponding disjunctive arcs are inserted in the graph. The resequencing procedure of the MSBH differs from the resequencing procedure of the original SBH. As in the original SBH, when all the machines are scheduled, the algorithm is stopped.

In the literature, branch and bound based algorithms have been used to solve L_{max} problems. For example, Adams et al. (1988) solve the single machine L_{max} problem by a version of Carlier's (1982) branch and bound algorithm. In Adams et al. (1988), at every node of the branch and bound tree, a heuristic based on the Most Work Remaining (MWKR) priority dispatching rule is applied to the current single machine problem. In this rule, the available operation belonging to the job with the largest amount of remaining processing time is selected as the next operation (i.e. largest sum of all unprocessed operation times).

In the single machine L_{max} problem, the objective is to minimize the L_{max}, and preemptions are not allowed. The proposed branch and bound procedure for this problem can be constructed as follows:

There is a single node at level 0, which is at the top of the tree. At this node none of the jobs have been put into any position in the sequence. There are *n* branches going down to *n* nodes at level 1. Each node at this level corresponds to a partial solution with a specific job in the first position of the schedule. So at each of these nodes, there are still *n-1* jobs whose positions in the schedule are yet to be determined. There are *n-1* arcs emanating from each node at level 1 to level 2. There are therefore *n x (n-1)* nodes at level 2. At each node at level 2, the two jobs in the first two positions are specified; at level *k* the jobs in the first *k* positions are specified.

It is often not necessary to consider every remaining job as a candidate for the next position. If at a node at level *k-1*, jobs $j_1,\ldots,j_{k-1}$ are assigned to the first *k-1* positions, job *c* has to be considered as a candidate for position *k* only if

$$r_c < \min_{l \in J}\left(\max\left(t, r_l\right) + p_l\right)$$

where *J* denotes the set of jobs not scheduled yet, *t* denotes the time the machine completes job j_{k-1} and is free to start the next job, p_l denotes the processing time of job *l*, and r_c and r_l denote the release date of job *c* and job *l*. If job *c* does not satisfy this inequality, then it makes sense to put the job that minimizes the right hand side in position *k* and job *c* in position *k+1*. This would not affect the completion time of job *c* in any way. So in this case job *c* does not have to be considered for position *k*. Several bounding schemes can generate lower bounds for nodes in the search tree. An easy lower bound for a node at level *k-1* can be established by scheduling the remaining jobs *J* according to the preemptive EDD rule. The preemptive EDD rule is known to be optimal for this particular problem when preemptions are allowed and thus provides a lower bound for the nonpreemptive schedule, then all nodes with a higher lower bound can be disregarded (Pinedo & Chao, 1999).

Solving a single machine L_{max} problem by a branch and bound method is time consuming. To show that, we illustrate its application for an example problem that is given in Pinedo & Chao (1999). Figure 4.1 shows the branch and bound tree of the problem.

Consider a single machine with the following four jobs:

Jobs	1	2	3	4
p_j	4	2	6	5
r_j	0	1	3	5
d_j	8	12	11	10

The objective is to minimize the maximum lateness L_{max}. Level 1 of the tree has four nodes: (1, * , * , *), (2, * , * , *), (3, * , * , *) and (4, * , * , *). Nodes (3, * , * , *) and (4, * , * , *) may be disregarded immediately, because job 3 is released at time 3; if job 2 would start its processing at time 1, job 3 still can start at time 3. Job 4 is released at time 5; if job 1 would start its processing at time 0, job 4 still can start at time 5.

Computing a lower bound for node (1, * , * , *) according to the preemptive EDD rule yields a schedule where job 3 is processed during the time interval [4,5], job 4 during the time interval [5,10], job 3 (again) during interval [10,15], and job 2 during interval [15,17]. The L_{max} of this schedule, which provides the lower bound for node (1, * , * , *), is 5. In a similar way a lower bound, which is 7, can be obtained for node (2, * , * , *).

According to the lower bound value, node (1,2, * , *) at level 2 can be selected for further investigation on this branch. The lower bound for this node is 6 which is determined by the (nonpreemptive) schedule 1,2,4,3. Proceed with node (1,3, * , *) at level 2. The lower bound is 5, and it is determined by the (nonpreemptive)

schedule 1,3,4,2. As the lower bound for node (1, ∗ , ∗ , ∗) is 5 and the lower bound for (2, ∗ , ∗ , ∗) is larger than 5, schedule 1,3,4,2 has to be optimal.

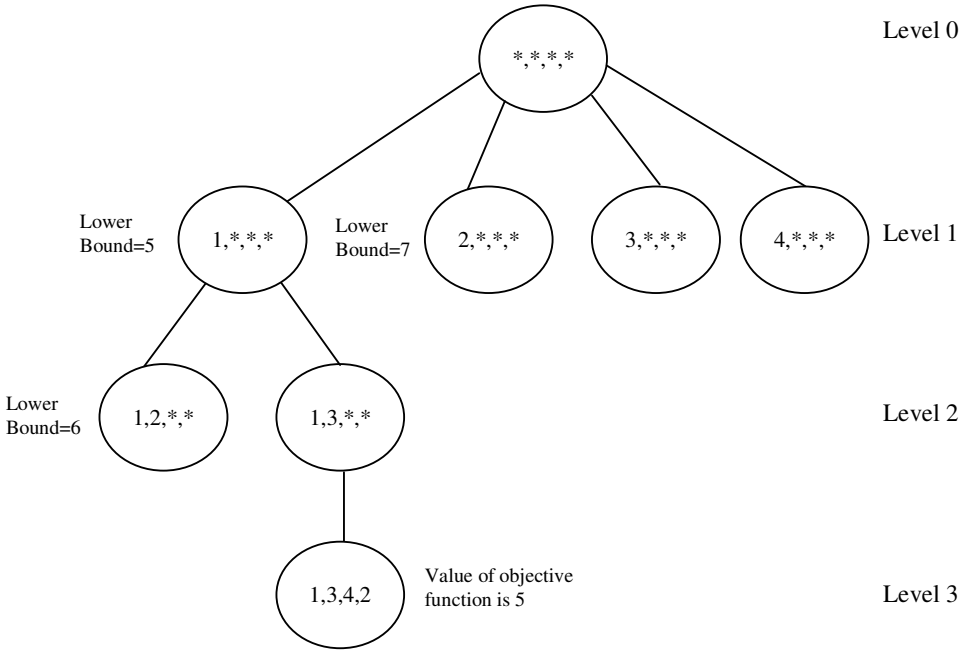


Figure 4.1 The branch and bound tree of the illustrative example

As can be discerned from the example, with the increase in the number of operations, the branches of the branch and bound tree will increase and the solution time will increase inordinately in order to reach an optimal schedule. In our application problem which will be described in detail in Chapter 7, there are approximately 100 operations that need to be scheduled on each machine throughout the day. For this reason the branches at each level will be numerous and our problem's being scheduled and solved by this method is beyond the capability and capacity.

As explained in the illustrative example, solution time of SBH in large-scale problems like ours can be excessively long. Thus, in our problem, the method which is used to solve the single machine problem in the original SBH should be modified. In the next section, the modified algorithm for the single machine L_{max} problem is described.

4.1.1 The Modified Algorithm For Solving The Single Machine Maximum Lateness Problem

Given that the scheduling problems in the real world possess hundreds of machines and jobs, it is required that a new method with shorter solution time and tolerable solution quality concurrently be developed. Therefore, in order to solve large-scale problems in relatively shorter time frames, a new algorithm based on sequencing has been developed for the L_{max} problem. The new algorithm is inspired by the Branch and Bound Algorithm described, and it is as follows:

The Modified L_{max} Algorithm:

Suppose i denotes the job, j denotes the operation and k denotes the machine, O_k denotes the set of operations to be scheduled on machine k, p denotes the processing time, r denotes the release date and t denotes the completion time.

For each single machine subproblem:

Iteration 1.

Step 1. Compute the release date plus processing time for all operations to be processed on that machine and find the minimum.

Step 2. Define the operations whose release dates are smaller than the minimum value found in Step 1.

Step 3. If there is only one operation such as defined in Step 2, schedule this operation as the first one.

If there is more than one operation, schedule the one with the minimum due date. (If there are ties, break them by giving preference to the smallest release date. If there are still ties, break them by giving preference to the smallest operation index.)

Step 4. Calculate the completion time for the scheduled operation.

$$t_{ijk} = r_{ijk} + p_{ijk} \; ; \; t = t_{ijk}$$

Iteration 2.

Step 1. (i) Find the operations that have not been scheduled yet with release dates either smaller or equal to the time value.

(ii) Find the operations that haven't been scheduled yet with release dates bigger than the time value.

Step 2. (i) If there is only one operation in Step 1 (i), schedule this.

(ii) If there is more than one operation in Step 1 (i), schedule the one with the minimum due date (if there are ties, break them by giving preference to the smallest release date. If there are still ties, break them by giving preference to the smallest operation index.)

(ii) If there is no operation in Step 1 (i), for operations in Step 1 (ii) apply Iteration 1.

Step 3. Calculate the completion time for the scheduled operation.

$$t_{ijk} = t + p_{ijk}\ ;\ t = t_{ijk}$$

Iteration 3.

Step 1. Repeat Iteration 2 until all operations of the machine are scheduled and record this operation sequence as resultant sequence for that machine.

Step 2. Calculate L_{max} for the resultant sequence on the associated machine.

$$L_{\max}(k) = \max_{(i,j)\in O_k} \{t_{ijk} - d_{ijk}\}$$

The machine with the largest L_{max} is determined after the aforementioned L_{max} algorithm is applied to all unscheduled machines at each iteration. In the event that there is more than one machine with the largest L_{max} value (breaking the tie), the machine which has the smallest index number is selected. The selected machine is the "bottleneck machine" of this iteration. The graph representing the partial schedule is updated by fixing the disjunctive arcs corresponding to the sequence of the operations on the bottleneck machine. The makespan (C_{max}) is increased by L_{max}*(bottleneck).*

$$C_{max}(iteration) = C_{max}(iteration\text{-}1) + L_{max}(bottleneck)$$

4.1.2 The Modified Resequencing Procedure

An important component of the SBH is the resequencing procedure. In particular, results show that the high quality solutions for the subproblems and extensive resequencing are essential to obtaining good solutions. In the MSBH the resequencing step differs from the original SBH of Adams et al. (1988).

In resequencing step, the sequence of operations on each machine that has been scheduled up to that point is recalculated successively while keeping the sequences fixed on all other scheduled machines. In this step, the maximum number of local resequencing cycles has been increased to six which is three in the original SBH. It means that, if an improvement is found after this step, the process is repeated up to a maximum of six times. During the first resequencing cycle, the machines are considered in the order in which they were introduced into the solution. After the first resequencing cycle, the machines are reordered in decreasing order of their L_{max} values at the end of each resequencing cycle.

In original SBH, once the aforementioned resequencing process is completed, it will be once again repeated after removing the last α noncritical machines from the solution. A machine is non-critical if no arc corresponding to an operation on the machine is on the critical path. Finally, the non-critical machines are reintroduced into the solution and the resequencing process is repeated. We have also cancelled the second part of the local resequencing procedure, where α noncritical machines are temporarily removed, because we have not found real improvement in doing so and it is time consuming. (Dauzere-Peres & Lasserre, 1993).

The resequencing procedure is clearly computationally intensive, involving many single machine problems and longest path calculations. To evaluate the impact of this step to the SBH performance, we consider two alternative resequencing procedures. In the first one, we perform the full resequencing step described above after a machine has been introduced into the schedule. In the second one, we perform

resequencing for different resequencing levels from one to six. Furthermore, we omit the resequencing step totally.

4.1.3 The Improved Cycle Elimination Procedure

In the SBH, it may be the case that an operation waiting to be scheduled on a machine must be processed after certain other operations have been completed on that machine because of the compulsory sequence restrictions on the operations. Between the processing of two operations subject to these precedence constraints, a certain minimum amount of time (that is, a delay) must elapse. These precedence constraints are delayed precedence constraints. Without these constraints the SBH may end up in a situation where there is a cycle in the disjunctive graph and the corresponding schedule is infeasible.

Suppose the example in Figure 4.2. If arc (2,1)→(2,2) would be inserted, then a cycle is created in the graph and the overall schedule is infeasible. This situation could have been prevented by imposing delayed precedence constraints. After scheduling machine 1, a path from (2,2) to (2,1) with length three is inserted to the graph. It means that, operation (2,2) must precede operation (2,1), and furthermore, there must be a delay of two time units between the completion of operation (2,2) and the start of operation (2,1).

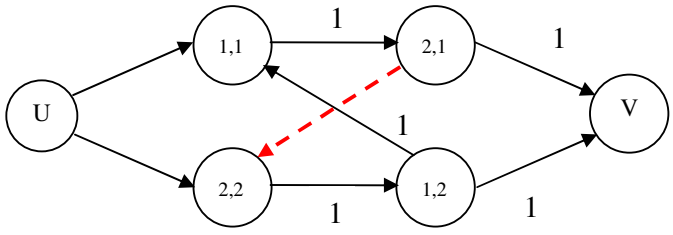


Figure 4.2 An infeasible graph

In the MSBH, a cycle elimination procedure based on delayed precedence constraints approach is developed to prevent infeasibility. It is proposed to assist in maintaining schedule feasibility during the MSBH. The cycle elimination procedure works as follows:

Suppose operation *j* is the last operation which was scheduled on machine *k* by the modified L_{max} algorithm, and the algorithm picks up the next operation to be scheduled on machine *k*.

Step 1. Find the predecessors of the operation *j* that were scheduled in the previous iterations of the MSBH.

Step 2. For each operation *l* which is found in Step 1;
Find the operations which use the same machine with operation *l* and are sequenced on the machine before operation *l*.

Step 3. For each operation *m* which is found in Step 2;
If one of the predecessors of operation *m* uses the machine *k* and is not scheduled yet, do not schedule it as the immediate next operation after the operation *j* on the machine *k*.

Consider the graph in Figure 4.2. The machine 1 is scheduled as (1,2)→(1,1). While the machine 2 is being scheduled as (2,1)→(2,2) in the next iteration, a cycle is created among these four operations and the overall schedule will be infeasible. So, a constraint that impedes scheduling operation (2,1) before operation (2,2) has been introduced in the MATLAB code of the MSBH, as described above. When the modified L_{max} algorithm is choosing the next operation to be scheduled, the operation that may cause a cycle is not added to candidate set. Thus, it prevents from forming a cycle. Furthermore, there is no need to insert an arc between (2,2)→(2,1) with 3 time units weight.

4.1.4 The Improved Method For The Reentrant Flow

In defining the operations sequence for each machine, the RJSSP and the classical JSSP fall apart because of the reentrant product flows. So, we insert reentrant properties in the MSBH.

In the RJSSP, there are conjunctive arcs instead of disjunctive arcs, among the operations that belong to the same job and use the same machine. While a machine is being scheduled, the preceding operation of a job must be scheduled before the succeeding operation of that job on the same machine. While the sequence of operations on the machine is being determined, the precedence relationships among these reentrant operations must be ensured. The release dates of the reentrant operations will shift because of the prerequisite operations among them. This will impact the solution of single machine L_{max} problem in the MSBH. Because, in the modified L_{max} algorithm, the computed time value for the operations that cause reentrant flow will change and this will impact the choice of the successive operation to be selected. For this reason, new constraints have been introduced to the MATLAB program for the reentrance situation. This is shown on the illustrative example given in Section 4.1. Table 4.1 shows the routing and Figure 4.3 shows the disjunctive graph of the example. The first iteration of the MSBH is explained as follows.

Table 4.1 The routing of the illustrative example

Jobs	Machine sequence	Processing times
1	2, 3, 1, 3, 1, 3	$p_{21} = 10$, $p_{31a} = 1$, $p_{11a} = 9$, $p_{31b} = 3$, $p_{11b} = 7$, $p_{31c} = 6$
2	1, 2, 3, 2, 3, 1	$p_{12a} = 5$, $p_{22a} = 6$, $p_{32a} = 8$, $p_{22b} = 9$, $p_{32b} = 2$, $p_{12b} = 3$
3	1, 3, 2, 1, 2, 1	$p_{13a} = 4$, $p_{33} = 4$, $p_{23a} = 5$, $p_{13b} = 1$, $p_{23b} = 5$, $p_{13c} = 7$

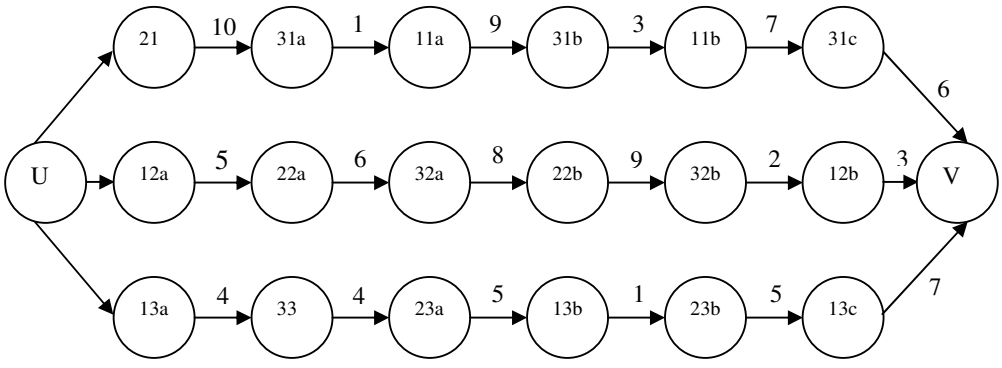


Figure 4.3 Disjunctive graph of the illustrative example

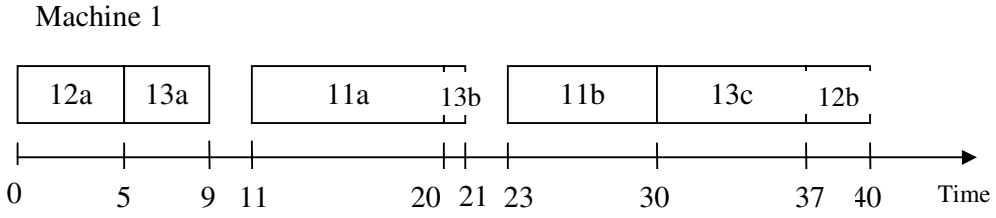
Iteration 1. The data for the single machine L_{max} problem corresponding to machine 1 is determined as follows:

Job	1		2		3		
Operation	11a	11b	12a	12b	13a	13b	13c
Processing time	9	7	5	3	4	1	7
Release date	11	23	0	30	0	13	19
Due date	20	30	8	36	14	24	36

The operations which lead to reentrant flows are as follows:

- Operations 11a and 11b belong to job 1 and use machine 1,
- Operations 12a and 12b belong to job 2 and use machine 1,
- Operations 13a, 13b and 13c belong to job 3 and use machine 1.

The sequence obtained by the proposed L_{max} algorithm in Section 4.1.1 turns out to be 12a→13a→11a→13b→11b→13c→12b with $L_{max}(1)=4$.



$L_{max}=max\{(5-8=-3),(9-14=-5),(20-20=0),(21-24=-3),(30-30=0),(37-36=1),(40-36=4)\}=4$.

How the improved method for the reentrant flow works is as follows.

Suppose operations 12a, 13a, 11a and 13b were scheduled successively via the proposed L_{max} algorithm, and the algorithm searches the next operation to be scheduled.

After scheduling operation 13b, in the proposed L_{max} algorithm, the algorithm picks up the unscheduled operation which has a release date smaller than the completion time of operation 13b [which is 21 time units]. So, operation 13c, that has a release date of 19 time units, is picked up to be scheduled in the next step.

Operations 13b and 13c belong to job 3. Because they belong to the same job (job 3) and use the same machine (machine 1), they cause reentrance for job 3. The release date of operation 13c is 19 time units; therefore normally it can start at time 21 which is the completion time of operation 13b. Yet, due to the precedence relationship between operations 13b and 13c; in order for operation 13c to start after operation 13b, operation 23b must be completed as a prerequisite. Operation 23b, which belongs to job 3 and uses machine 2, is the successor of operation 13b and the predecessor of operation 13c. So, operation 13c can not start before operation 23b is finished. The processing time for operation 23b is 5 time units as it is seen in Table 4.1. So, operation 13c can be started at time 21 + 5 = 26 rather than at time 21. The algorithm searches thoroughly if any of the unscheduled operations has a smaller release date than 26 time units with a tightness due date with respect to current release dates because of the reentrant flows. So, operation 11b which has a release date of 23 time units and a due date of 30 time units is selected as the next operation instead of operation 13c.

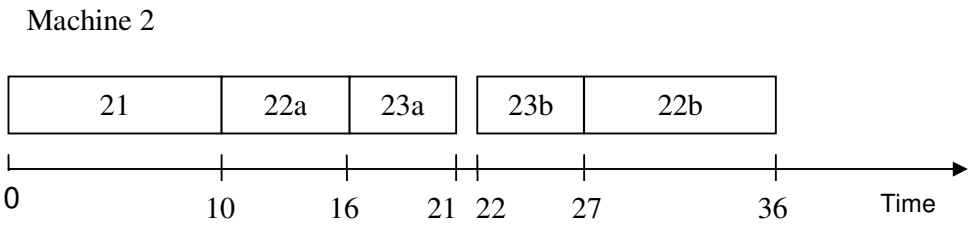
The data for the subproblem regarding machine 2 are as follows:

Job	1	2		3	
Operations	21	22a	22b	23a	23b
Processing time	10	6	9	5	5
Release date	0	5	19	8	14
Due date	10	14	31	23	29

The operations which cause reentrant flows are as follows:

- Operations 22a and 22b belong to job 2 and use machine 2.
- Operations 23a and 23b belong to job 3 and use machine 2.

The sequence obtained by the modified L_{max} algorithm turns out to be 21→22a→23a→23b→22b with $L_{max}(2)=5$.



$L_{max} = max$ {(10-10=0),(16-14=2),(21-23=-2),(27-29=-2),(36-31=5)} = 5.

Operations 23a and 23b belong to job 3. The release date of operation 23b is 14 time units; therefore normally it can start at time 21 which is the completion time of operation 23a. Yet, due to precedence relationship in order for operation 23b to start after operation 23a, operation 13b must be completed as a prerequisite. Operation 13b, which belongs to job 3 and uses machine 1, is the successor of operation 23a and the predecessor of operation 23b. Processing time for operation 13b is 1 time units as it seen in Table 4.1. So, operation 23b starts at time 21+1=22 rather than at time 21. The algorithm searches thoroughly if any of the unscheduled operations has a smaller release date than 22 time units with a tightness due date with respect to current release dates because of the reentrant flows. But, there is not. So, operation 23b is scheduled as the next operation.

The data for the subproblem regarding machine 3 are as follows:

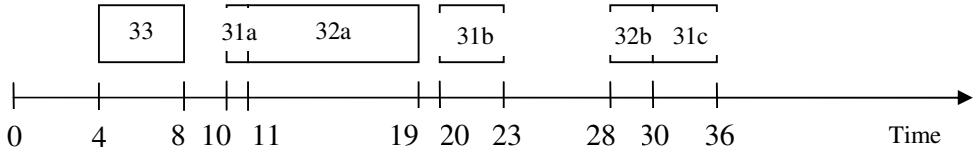
Job	1			2		3
Operations	31a	31b	31c	32a	32b	33
Processing time	1	3	6	8	2	4
Release date	10	20	30	11	28	4
Due date	11	23	36	22	33	18

The operations which cause reentrant flows are as follows:

- Operations 31a, 31b and 31c belong to job 1 and use machine 3.
- Operations 32a and 32b belong to job 2 and use machine 3.

The sequence obtained by the modified L_{max} algorithm turns out to be 33, 31a, 32a, 31b, 32b, 31c with $L_{max}(3)=0$.

Machine 3



$L_{max}=max\{(8-18=-10),(11-11=0),(19-22=-3),(23-23=0),(30-33=-3),(36-36=0)\}=0$

Operations 31a, 31b, 31c are all processed on machine 3 and belong to job 1. Operations 32a, 32b are both processed on machine 3 and belong to job 2. But, no delay on machine 3 is experienced due to reentrance. Because the elapsed time between these operations which belong to the same job is enough to ensure the precedence relationships.

The sample MATLAB codes of the MSBH with makespan minimization objective for the RJSSP are given in Appendix A. Furthermore, this improved heuristic is tested on various sizes of test problems in Chapter 6.

In the next section, the TML and TOTINF subproblem criticality measures that are adapted to the MSBH are described.

4.2 The Modified Shifting Bottleneck Heuristic With Total Machine Load And Total Infeasibility Subproblem Criticality Measures

If there are numerous machines and operations, it requires a long processing time to determine the bottleneck machine in the SBH. Therefore, there is need for another method which depletes less time to determine the bottleneck machine in each iteration among the subproblems of the SBH. The literature review has been conducted and it has been determined that TML and TOTINF methods which are proposed by Aytug et al. (2002) are suitable for the SBH with the makespan objective. As a result, the solution time of the SBH can be reduced with negligible loss in solution quality.

4.2.1 Total Machine Load

The static measures of subproblem criticality can be calculated rapidly using job and machine attributes at the start of the procedure. The subproblems (machines) are ordered in decreasing order of these indices and this order does not change throughout the SBH. Thus, some time savings are expected since there is no longer a need to solve all unsolved subproblems optimally at each iteration. Instead, only the subproblem with highest criticality needs to be solved optimally.

TML uses the total workload on each machine, which is given by the sum of the processing times of all operations of the jobs processed on that machine. The intuition here is that the machine where jobs require long processing times is likely to be a bottleneck, in the sense that poor scheduling decisions on this machine are

likely to lead to a poor schedule for the overall procedure. Note that this is also a proxy for the machine utilization over the period considered in the schedule.

4.2.2 Total Infeasibility

The SBH tries to find an optimal solution by increasing the objective value as little as possible at each iteration while progressively enforcing the machine capacity constraints. Since all unscheduled machines are assumed to have infinite capacity, at any partial schedule there may well be a number of machines whose capacity constraints are violated. It means that these machines are required to process more than one job during a given time interval. Focusing on the subproblem that has the 'highest' level of infeasibility and imposing the capacity constraints for this subproblem by scheduling it optimally might well improve the solution quality and time of the SBH.

In order to evaluate the degree of infeasibility at a given machine in any partial solution, the number of jobs processed on each unscheduled machine in each time interval is calculated by using the release and due dates. If this number is greater than one, the schedule is infeasible since each machine can process only one job at a time. An infeasibility profile for each machine over time is achieved by the result of this calculation. Consider the illustrative example given in Section 4.1. The Gannt chart for release and due dates of operations which are processed on machine 1 is given in Figure 4.4. The infeasibility profile of machine 1 is given in Figure 4.5. The TOTINF is given by the area under the profile for the intervals, where there is more than one job being processed. Consequently, it is a lot faster than solving each subproblem optimally.

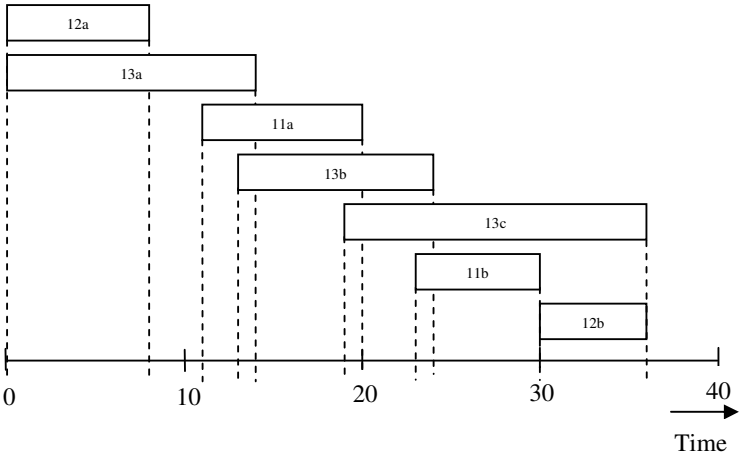


Figure 4.4 The Gannt chart of machine 1

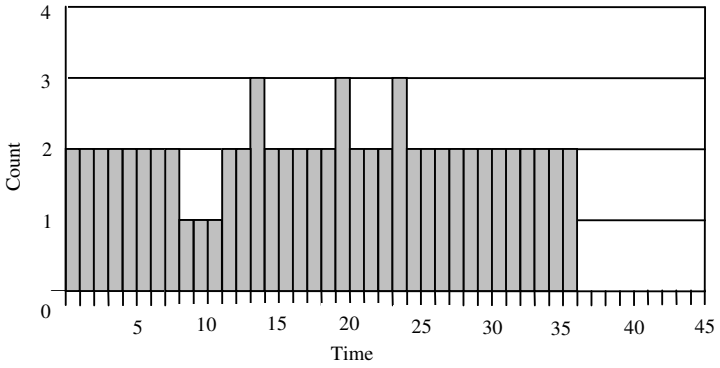


Figure 4.5 The infeasibility profile of machine 1

In this study, we used TML and TOTINF measures, which are good at makespan minimization problem, for the order in which to solve the subproblems in the MSBH. Our aim is to reduce the central processing units (CPU) requirements of the MSBH without compromising the quality of the solutions obtained. Because our application problem is a large job shop, there will be no need to solve all the subproblems which have many operations.

To summarize, in the MSBH with TML, while the bottleneck machine is being determined, rather than solving L_{max} problem for all unscheduled machines at each iteration of the SBH, TML values of machines are calculated before starting to solve the problem. On the other hand, in the MSBH with TOTINF, TOTINF for all unscheduled machines at the outset of each iteration is calculated and the machine

with the maximum TOTINF is considered as the bottleneck rather than solving L_{max} problems for all unscheduled machines at that iteration.

In our test problems as described in Chapter 6, TML and TOTINF measures yield comparable performance with the dynamic measure used in the MSBH.

In the next chapter, we describe the modified Rolling Horizon Procedure (MRHP) proposed for large job shops. We explain the proposed approach which is used instead of overlapping in the MRHP for makespan minimization objective. Furthermore, we describe the improved rule for assignment of operations to time windows which will be used to solve the application JSSP in the textile factory.

CHAPTER FIVE
THE MODIFIED ROLLING HORIZON PROCEDURE

In the previous chapter, the modified Shifting Bottleneck Heuristic (MSBH) for reentrant job shops was described. In this chapter, a modified Rolling Horizon Procedure (MRHP) is developed for large reentrant job shops, in which the makespan must be minimized. The method divides the problem into a number of subproblems each corresponding to a time window of the overall schedule. These subproblems are solved using the MSBH with Total Machine Load (MSBH_TML) as described in Chapter 4.

Furthermore, a number of rules for assigning operations to time windows are derived. In addition, we develop a modified decomposition procedure for a special job shop which is the application problem of this study in a textile plant. The proposed method is tested by using instances up to 20 machines and 100 jobs in the next chapter.

5.1 The Modified Rolling Horizon Procedure

In the job shop scheduling problem (JSSP), as the number of operations and machines in the problem increase, the processing time required for scheduling increases as well. With respect to machine based decomposition applied with Shifting Bottleneck Heuristic (SBH), in order to reduce the operation and machine quantity of each iteration, it will be suitable to break the problem network into pieces based on time. In Rolling Horizon Procedure (RHP), the first decomposition is done horizontally in the Gantt chart using time windows, and the second decomposition is done vertically using the machine decomposition of the SBH.

When the literature is reviewed, for the makespan objective function, applying RHP would not be as efficient as that for the tardiness case. Because, the total weighted tardiness function is associated with each operation, while the makespan depends on a combination of all the operations in a simultaneous manner. Therefore,

a new methodology has been required for the makespan minimization objective function.

Upon review, to reveal effective results with RHP for makespan objective, when updating release dates and due dates in the SBH, not only the relevant time window but the entire graph must be taken into consideration. Therefore, after a machine scheduled in a time window, release dates and due dates of all operations in disjunctive graph are required to be revised. In respect of necessity, we modified the RHP which is proposed by Singer (2001) for the makespan objective. Figure 5.1 shows the general scheme of the MRHP.

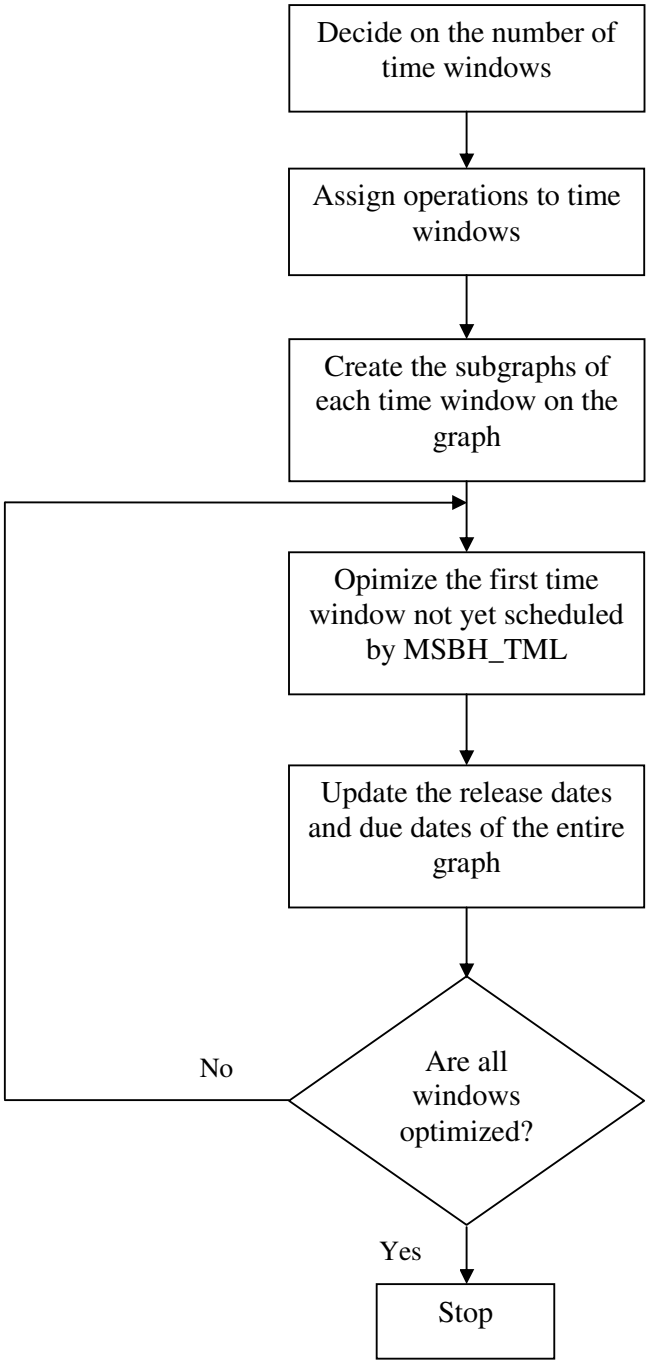


Figure 5.1 General scheme of the MRHP

In general, as described in Section 3.3, three decisions are involved in a RHP:

- Assignment of operations to time windows.
- Optimization of each time window.
- Overlapping degree.

For the MRHP, the proposed rules for assigning operations to time windows and the proposed procedure to optimize each time window are described in the remainder of the section. Furthermore, to prevent overlapping, a new method is proposed.

First of all, to understand the effect of overlapping, consider the instance with 4 machines and 4 jobs for a classical job shop that is given in Table 5.1.

Table 5.1 The data of the instance

Jobs	Machine sequence	Processing times
1	3, 1, 4, 2	80, 80, 65, 55
2	1, 3, 4, 2	180, 35, 65, 45
3	4, 3, 1, 2	90, 35, 145, 60
4	2, 3, 1, 4	110, 110, 95, 35

The graph corresponding to the instance (Figure 5.2) is divided into two time windows. The original SBH is applied to the first time window. Figure 5.3 is the graph depicting the first time window after being scheduled. Figure 5.4 is a Gantt diagram representing the schedule obtained for the first time window.

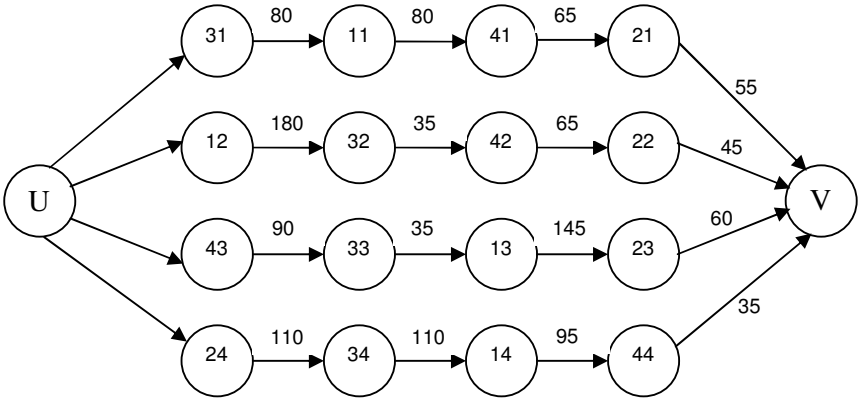


Figure 5.2 Overall graph for the instance

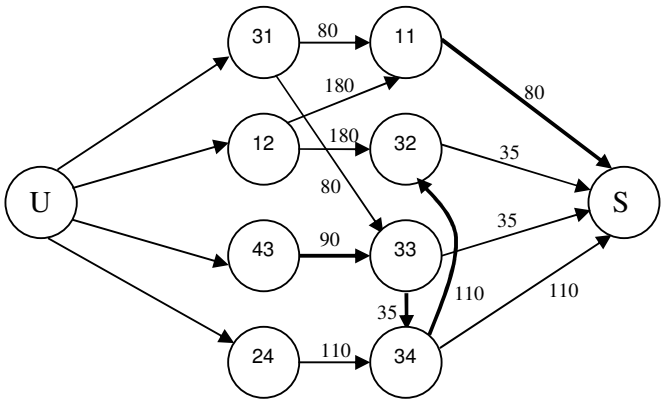


Figure 5.3 Graph corresponding to the first time window

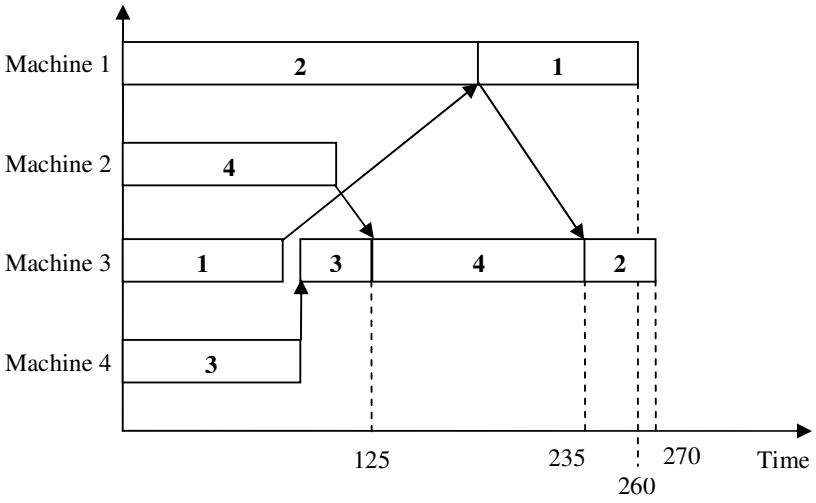


Figure 5.4 Gantt diagram for the first time window of the problem

The transition nodes between the first and the second windows are (1,1), (3,2), (3,3) and (3,4). As it is seen in Figure 5.4, their completion times in the first time window are 260, 270, 125 and 235 respectively. These values, as shown in Figure 5.5, will be used to initialize the graph of the second window. The release time of node (1,3) is the completion time of node (3,3) and so on. Then, the original SBH is applied to the second time window.

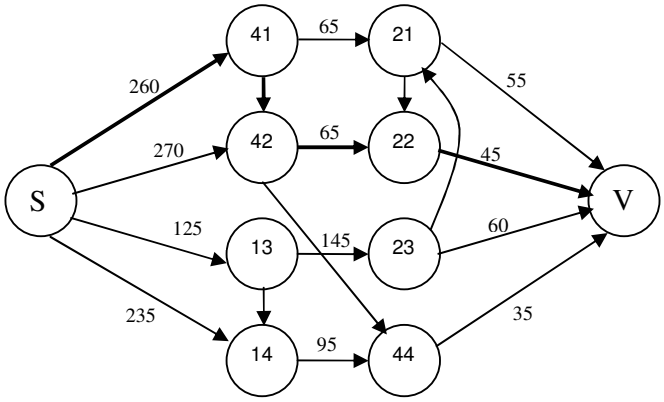


Figure 5.5 Graph associated to the second time window

The value of the critical path at this stage is 435. All nodes are scheduled, but the solution may be inconsistent (Figure 5.6). For instance, node (1,3) is supposed to be processed at t = 125 on machine 1, while this machine is not available before t = 260 (the completion time of node (1,1) in the first window). As a consequence, if the algorithm stops here, the critical path in this graph will be shorter than the real critical path of the problem.

To deal with the overlapping problem depicted in Figure 5.6, the graph should be updated. As it is seen in Figure 5.5, three machines are involved in this time window. The first node scheduled on machine 1 is node (1,3). The incoming arc of this node (Figure 5.5) has a current value of 125. This value means that job 3 is ready to be scheduled, but it does not guarantee that the machine on which it has to be scheduled (machine 1) is ready.

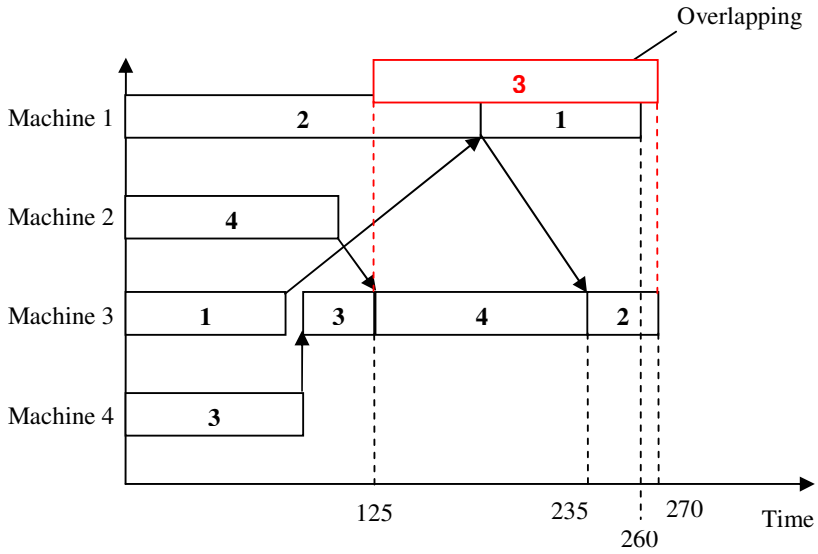


Figure 5.6 Gantt diagram showing the overlapping of operations 11, 12 and 13 in the second time window

The completion time of machine 1 in the first time window is greater than the completion time of the last operation of job 3 done in the first time window (node (3,3)). Therefore, the weight of the incoming arc of node (1,3) is updated so that the completion time of node (1,1) is considered. The new weight of the arc linking the source and node (1,3) becomes 260. After this update, a realistic solution is obtained as depicted in the Gantt diagram of Figure 5.7 which presents the final schedule. As a result, the makespan of the problem is 565 time units.

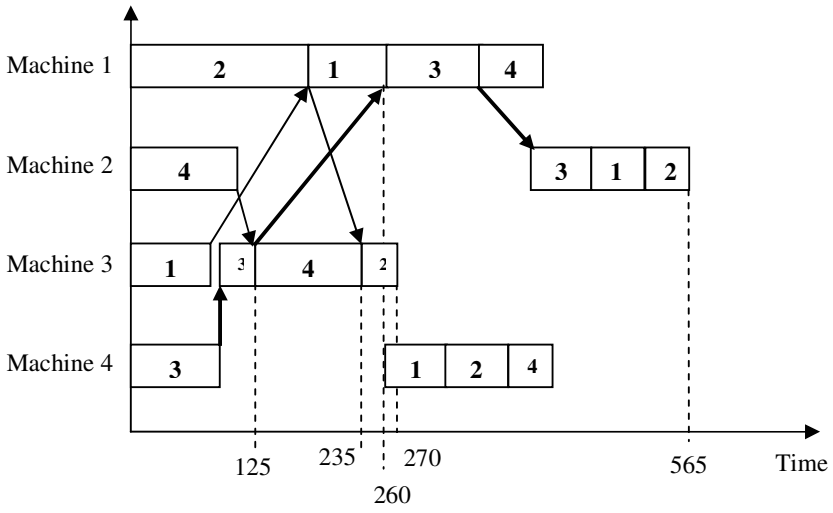


Figure 5.7 Gantt diagram of the updated graph

5.1.1 Assignment Of Operations To Time Windows

In a RHP, after determining the number of time windows, the operations that are assigned to each time window is designated. In the literature, we have not come across a method used for determining the number of time windows, but rather have seen interpretations that state it would be proper for time window quantity fall in between 2-7 depending on the size of the problem. It is worth noting that as the window quantity increases, the problem will divert from the optimal. In this study, the number of time windows differs from 2 to 5, according to the problem size, and it is the input for the MRHP that has been coded in MATLAB. Several heuristics can be studied and developed to assign operations to time windows. For instance, assigning the first "x" operations of each job – x being a decision variable, to the first time window. Other more complex rules can also be derived. In this study, the rules for assignment of operations to time windows which are given in Singer (2001) are used. However, we modified the third rule which is the time window with a balanced workload for each job. These rules can be summarized as follows:

The first rule is time window with a fixed number of operations per job. Initially, the maximum of the total number of operations per job is determined. Then, this number is divided into the number of time windows used. The calculated value

will be the number of operations per job in each time window. After that, operations are assigned to each time window by this fixed number successively until all the operations of the jobs are assigned.

The second rule is time window with a balanced number of operations per machine. First of all, the maximum of the total number of operations per machine is determined. Then, this value is divided into the number of time windows which is denoted by x. The assignment procedure begins with assigning the first operation of each job to the first time window and assigning the next operation of each job while ensuring precedence relationships. When the number of operations assigned to each machine is equal to x in a time window, it passes to the next time window.

The third rule is time window with a balanced workload for each job. First, the maximum of the total processing times of jobs is determined. Then, it is divided into the number of time windows, denoted by x. Beginning from the first operation of each job, operations are assigned to time window until the workload of the job in a time window is equal to x. The following windows are created in the same way by unassigned operations.

5.1.2 Optimization Of Each Time Window

In the MRHP, the general approach that will be adopted to solve the reentrant job shop scheduling problem (RJSSP) with makespan objective is as follows: Instead of considering different subgraphs for each time window of the problem, only one general graph is considered. The optimization of each time window is done on the general graph. In Singer (2001), different subgraphs are used for each time window, and the different time windows are partially decoupled by using overlapping.

The algorithm is as follows:

Step 1.Machines corresponding to the first time window will be scheduled using the MSBH_TML. When a machine is added, the entire graph is updated, and resequencing of all the machines scheduled earlier is done.

Step 2. In the other time windows, while some machines are not scheduled yet:

Select the bottleneck machine in the current window using the MSBH_TML.

Add the corresponding arcs to the graph.

Check if this machine is existent in the previous time windows. If this is the case, link the first node to be scheduled on this machine in this window to the last node scheduled on this machine in the previous window. This approach prevents overlapping.

Update the entire graph.

Resequence all the machines scheduled earlier.

To clarify that the approach guarantees the absence of overlapping, consider the example given in Table 5.1. The assignment of operations to time windows is done by the fixed number of operations per job rule. The first time window is scheduled. While scheduling the second time window, using the original SBH, assume machines 3 and 4 are scheduled, and following them, machine 1 is scheduled in the second time window, so that operation (1,3) is done before operation (1,4). Two arcs will be added: one, linking (1,3) to (1,4) and the other linking (1,3) to the last operation done on machine 1 in the previous time window that is node (1,1) as it is seen in Figure 5.8. Thus, no overlapping is occurred.

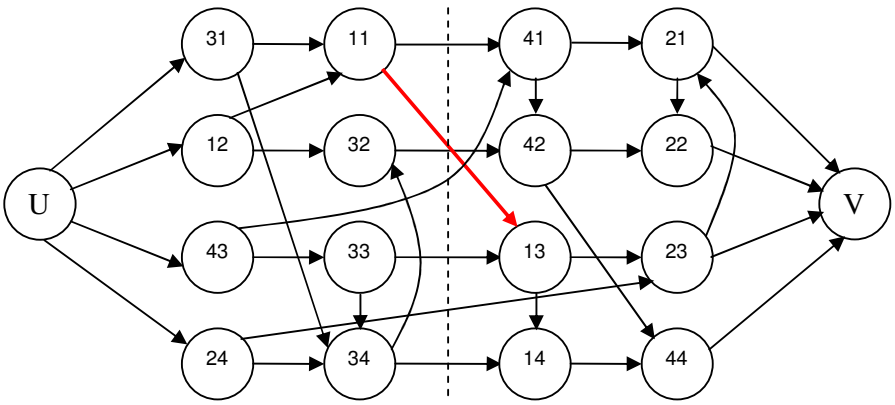


Figure 5.8 Updating the nodes in the second time window

In summary, in the MRHP, rather than using different subgraphs for each time window, optimization of each time window is conducted on a general graph. Machines of each time window are scheduled by the MSBH. To achieve process simplification, while determining bottleneck machine at each iteration, TML subproblem criticality measure is used. Each time a machine is scheduled, corresponding arcs are installed to the graph and the entire graph is updated.

When skipped to next time window, the bottleneck machine is determined via the MSBH_TML. New arcs that belong to this machine are installed to the graph. Whether this machine has been scheduled or not in previous time windows is checked. If scheduled, the last operation scheduled on the machine in the previous time window and the first operation scheduled on the machine in this time window are linked by an arc. It prevents the schedule for one time window from interfering with the next one. By this approach, overlapping is avoided between consecutive time windows. Due to the feature of makespan objective, an update is done after adding any machine, and this update considers the whole graph, not just the nodes to be scheduled in the current time window. For makespan objective, this method is more precise than the RHP given in Singer (2001).

5.2 Improved Decomposition Heuristic For A Special Job Shop

In the previous section, the RHP which is presented by Singer (2001) is modified for makespan minimization objective. In this section, we propose a new rule for the assignment of operations to time windows, because the application job shop which is the subject of this thesis has a unique character. All jobs that come to the shop are processed by the rolling machine first. In order to process fabrics on the other machines, they must be rolled in to the docs first. After being processed on several machines, their last operation is performed on the fabric control machine. Afterward, they get ready for shipment. Therefore, the first and last operations of the jobs are performed on the same machines. This case renders the job shop a unique character.

As a matter of fact, to prevent job accumulation within the time windows because of the unique character of the job shop, a different methodology is required to assign operations to time windows. So, "balanced machine load" methodology has been developed. This means to balance the total processing time of all machines (TML gives total processing times of operations that are processed on a machine) at each time window. The proposed procedure begins with assigning the first operation of the first job to the first time window. It continues to assign operations as much as the time window TML value for each machine is filled up. Therefore, operations to be assigned to first time window are determined. Operations to be assigned to other time windows are determined likewise.

In the original SBH, the machine with the maximum L_{max} is considered as the bottleneck. In the MSBH_TML, the machine with the maximum TML is chosen as bottleneck. If the TML of a machine in a time window is higher than the other machines' TML values, the lateness values of the operations processed on that machine will be high. Thus, L_{max} value will be high and this will increase makespan. Therefore, each time window must be a smaller scale model of the network of the problem, so that no excessive makespan increase is experienced. As a result, when determining the operations to be assigned to time windows, machines are assigned in a way to balance TML values in each time window. Moreover, to prevent job

accumulation in the shop and to avoid makespan extensions of short jobs, jobs with less total processing times are sequenced first whereas the other are sequenced after. So that, a Shortest Processing Time dispatching rule based approach, which yields good results for JSSPs in literature, has been used. The objective is scheduling the short duration jobs first and therefore to shorten the stay period of jobs in the shop and to prevent to grow the L_{max} and so makespan.

The developed algorithm is comprised of the following six steps.

Step 1. Number of time windows is defined with respect to the scale of the problem.

Step 2. Total processing times of the jobs are defined.

Step 3. The jobs are sequenced in an ascending manner based on total processing times.

Step 4. Operations are assigned to windows according to balanced machine load rule.

Step 5. Beginning from the first time window, operations are scheduled to machines via the MSBH_TML, respectively.

After a machine is scheduled, the first operation scheduled to that machine in this time window and the last operation scheduled to the machine in the previous time windows are linked by an arc. Thus, overlapping is prevented. The whole graph is updated.

Step 6. If all time windows are scheduled, the algorithm stops.

In this chapter, the MRHP for the RJSSP with makespan objective has been described in detail. In the next chapter, the proposed methods, the MSBH and the MRHP, that are explained in this and the previous chapters are tested on various randomly generated test problems. The results are compared with each other and with optimal results.

CHAPTER SIX
COMPUTATIONAL STUDY

In the previous chapters the proposed methods have been described to solve the reentrant job shop scheduling problems (RJSSP). In this chapter, these methods are tested on various randomly generated test problems and benchmark problems from the literature. The aim of this computational study is to see the effectiveness of the proposed methods in comparison to the original methods, and to compare the methods among themselves. Furthermore, to see the effectiveness of these models, the solution qualities and times are compared for different rescheduling levels and for different number of time windows.

In the following sections, test problem generation procedures, test results for small reentrant job shops, large reentrant job shops and the special reentrant job shop which is the application problem of this study are given successively. Furthermore, the comparison of the modified Shifting Bottleneck Heuristic (MSBH) and the original Shifting Bottleneck Heuristic (SBH) on the classical job shop scheduling problem (JSSP) is given for various benchmark problems from the literature.

6.1 Experimental Framework

A popular way to investigate the performance of mathematical formulations and heuristics is through computational study. In general terms, there are two opinions on the kind of instances to be used in testing models: random generation, and standard libraries. In our experimentation, both randomly generated test problems and benchmark problems from the literature are used. Furthermore, a real world problem is used which is described in the next chapter.

To assess the performance of the proposed methods discussed in the previous sections, we have conducted a series of computational experiments. The problems are static scheduling instances where all jobs are available simultaneously and all machines are idle at the start of the schedule. The objective in these problems is to

minimize makespan—there are no due dates. The proposed methods were implemented in MATLAB 6.0, and they were tested by solving a wide range of RJSSPs with various sizes and levels of hardness. The performance of the proposed methods has been evaluated with respect to two measures: the quality of solution and computational time.

Test problems are generated in MATLAB. Processing times, number of operations and machine sequences all follow discrete uniform distribution. The parameters for the test problems are the number of machines and jobs. The number of machines varies from 3 to 20. The number of jobs varies from 3 to 100 jobs and 4 to 50 operations per job. A total of 380 problems were tested using 30 different parameter sets for generating the problems. The computational tests are performed on a Pentium IV 1.7 GHz with 512 MB of RAM, measuring the time in central processing unit (CPU) time in seconds.

The computational experiment is divided into four categories. In the first section, the performance of the modified single machine maximum lateness (L_{max}) algorithm of the MSBH and the original SBH developed by Adams et al.(1988) is compared. Benchmark problems for classical JSSP from the literature are taken into consideration for this comparison. In the following sections, the results of randomly generated test problems for small and large reentrant job shops and the special reentrant job shop which has the properties of the application problem of this study are evaluated successively. In the set of experiments, the effects of rescheduling on the MSBH performance are explored. Furthermore, different numbers of time windows and different rules for assignment of operations to time windows are tested for the modified rolling horizon procedure (MRHP).

6.2 Comparison Of The Modified Shifting Bottleneck Heuristic With The Original Shifting Bottleneck Heuristic For The Classical Job Shop Scheduling Problem

The MSBH is tested on the set of benchmark problems generated by Lawrence (1984) [LA1-LA40] for classical job shops. The data set consists of 40 makespan (C_{max}) minimization problems. These problems are of four types: 10 job x 5 machine (LA1-LA5); 15 job x 5 machine (LA6-LA10); 20 job x 5 machine (LA11-LA15); 10 job x 10 machine (LA16-LA20); 15 job x 10 machine (LA21-LA25); 20 job x 10 machine (LA26-LA30); 30 job x 10 machine (LA31-LA35); 15 job x 15 machine (LA36-LA40). Five instances of each problem type are solved. Each job requires an operation on each machine and has an integer processing time.

The original SBH results used in the comparison have been obtained by FORTRAN on a VAX 780/11 by Adams et al. (1988). The optimal results are obtained from Pezzella & Merelli (2000).

6.2.1 Computational Results For Benchmark Problems

The optimal solutions of these benchmark problems, the results obtained by the original SBH and by the MSBH are given in Table 6.1. Table 6.2 represents the comparisons, and the summary of the results are shown in Table 6.3. The comparisons are made based on ratios of the solution values of the methods to optimum. Relative errors (RE(%)) show the solution gap between the methods.

Table 6.1 The results of the methods for LA1-LA40 instances

	optimum	**SBH**		**MSBH**	
	C_{max}	**C_{max}**	**time**	**C_{max}**	**time**
LA1	666	666	1.26	666	9
LA2	655	720	1.69	781	9
LA3	597	623	2.46	654	10
LA4	590	597	2.79	644	9
LA5	593	593	0.52	621	9
LA6	926	926	1.28	926	9
LA7	890	890	1.51	890	10
LA8	863	868	2.41	869	9
LA9	951	951	0.85	951	10
LA10	958	959	0.81	958	10
LA11	1222	1222	2.03	1229	11
LA12	1039	1039	0.87	1039	11
LA13	1150	1150	1.23	1150	11
LA14	1292	1292	0.94	1292	10
LA15	1207	1207	3.09	1207	10
LA16	945	1021	6.48	1064	14
LA17	784	796	4.58	836	14
LA18	848	891	10.20	908	14
LA19	842	875	7.40	976	14
LA20	902	924	10.20	1044	14
LA21	1046	1172	21.90	1171	19
LA22	927	1040	19.20	1059	21
LA23	1032	1061	24.60	1163	19
LA24	935	1000	25.50	1057	19
LA25	977	1048	27.90	1065	18
LA26	1218	1304	48.50	1340	37
LA27	1235	1325	45.50	1352	24
LA28	1216	1256	28.50	1312	26
LA29	1195	1294	48.00	1314	25
LA30	1355	1403	37.80	1419	25
LA31	1784	1784	38.30	1788	41
LA32	1850	1850	29.10	1850	42
LA33	1719	1719	25.60	1736	44
LA34	1721	1721	27.60	1796	42
LA35	1888	1888	21.30	1902	63
LA36	1268	1351	46.90	1399	57
LA37	1397	1485	61.40	1572	61
LA38	1196	1280	57.70	1403	66
LA39	1233	1321	71.80	1366	56
LA40	1222	1326	76.70	1462	46

Table 6.2 Comparisons of the methods for LA1-LA40 instances

	SBH - optimum comparison		MSBH - optimum comparison		MSBH - SBH comparison			
	C_{max}		C_{max}		C_{max}		time	
	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
LA1	1.000	0.00	1.000	0.00	1.000	0.00	7.143	614.29
LA2	1.099	9.92	1.192	19.24	1.085	8.47	5.325	432.54
LA3	1.044	4.36	1.095	9.55	1.050	4.98	4.065	306.50
LA4	1.012	1.19	1.092	9.15	1.079	7.87	3.226	222.58
LA5	1.000	0.00	1.047	4.72	1.047	4.72	17.308	1630.77
LA6	1.000	0.00	1.000	0.00	1.000	0.00	7.031	603.13
LA7	1.000	0.00	1.000	0.00	1.000	0.00	6.623	562.25
LA8	1.006	0.58	1.007	0.70	1.001	0.12	3.734	273.44
LA9	1.000	0.00	1.000	0.00	1.000	0.00	11.765	1076.47
LA10	1.001	0.10	1.000	0.00	0.999	-0.10	12.346	1134.57
LA11	1.000	0.00	1.006	0.57	1.006	0.57	5.419	441.87
LA12	1.000	0.00	1.000	0.00	1.000	0.00	12.644	1164.37
LA13	1.000	0.00	1.000	0.00	1.000	0.00	8.943	794.31
LA14	1.000	0.00	1.000	0.00	1.000	0.00	10.638	963.83
LA15	1.000	0.00	1.000	0.00	1.000	0.00	3.236	223.62
LA16	1.080	8.04	1.126	12.59	1.042	4.21	2.160	116.05
LA17	1.015	1.53	1.066	6.63	1.050	5.03	3.057	205.68
LA18	1.051	5.07	1.071	7.08	1.019	1.91	1.373	37.25
LA19	1.039	3.92	1.159	15.91	1.115	11.54	1.892	89.19
LA20	1.024	2.44	1.157	15.74	1.130	12.99	1.373	37.25
LA21	1.120	12.05	1.120	11.95	0.999	-0.09	0.868	-13.24
LA22	1.122	12.19	1.142	14.24	1.018	1.83	1.094	9.38
LA23	1.028	2.81	1.127	12.69	1.096	9.61	0.772	-22.76
LA24	1.070	6.95	1.130	13.05	1.057	5.70	0.745	-25.49
LA25	1.073	7.27	1.090	9.01	1.016	1.62	0.645	-35.48
LA26	1.071	7.06	1.100	10.02	1.028	2.76	0.763	-23.71
LA27	1.073	7.29	1.095	9.47	1.020	2.04	0.527	-47.25
LA28	1.033	3.29	1.079	7.89	1.045	4.46	0.912	-8.77
LA29	1.083	8.28	1.100	9.96	1.015	1.55	0.521	-47.92
LA30	1.035	3.54	1.047	4.72	1.011	1.14	0.661	-33.86
LA31	1.000	0.00	1.002	0.22	1.002	0.22	1.070	7.05
LA32	1.000	0.00	1.000	0.00	1.000	0.00	1.443	44.33
LA33	1.000	0.00	1.010	0.99	1.010	0.99	1.719	71.88
LA34	1.000	0.00	1.044	4.36	1.044	4.36	1.522	52.17
LA35	1.000	0.00	1.007	0.74	1.007	0.74	2.958	195.77
LA36	1.065	6.55	1.103	10.33	1.036	3.55	1.215	21.54
LA37	1.063	6.30	1.125	12.53	1.059	5.86	0.993	-0.65
LA38	1.070	7.02	1.173	17.31	1.096	9.61	1.144	14.38
LA39	1.071	7.14	1.108	10.79	1.034	3.41	0.780	-22.01
LA40	1.085	8.51	1.196	19.64	1.103	10.26	0.600	-40.03

Table 6.3 Summary of the comparisons for LA1-LA40 instances

	SBH - optimum comparison		MSBH - optimum comparison		MSBH - SBH comparison			
	C_{max}		C_{max}		C_{max}		time	
	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
avg.	1.036	3.58	1.070	7.04	1.033	3.30	3.756	275.63
min.	1.000	0.00	1.000	0.00	0.999	-0.10	0.521	-47.92
max.	1.122	12.19	1.196	19.64	1.130	12.99	17.308	1630.77

Analysis of Table 6.3 shows that, on the average, the benchmark problems are solved to within 3.58% of optimality by the original SBH and 7.04% of optimality by the MSBH. The MSBH finds the same objective function value in 9 of the 40 instances with the original SBH.

When we compare the results of the MSBH and the original SBH, the solution quality gap varies between -0.10% to 12.99%. The average solution gap is 3.30% for the set of benchmark problems. We prove that the MSBH yields near solutions in average when compared with the original SBH. Figure 6.1 shows the solution quality ratios of the instances for the MSBH and the SBH.

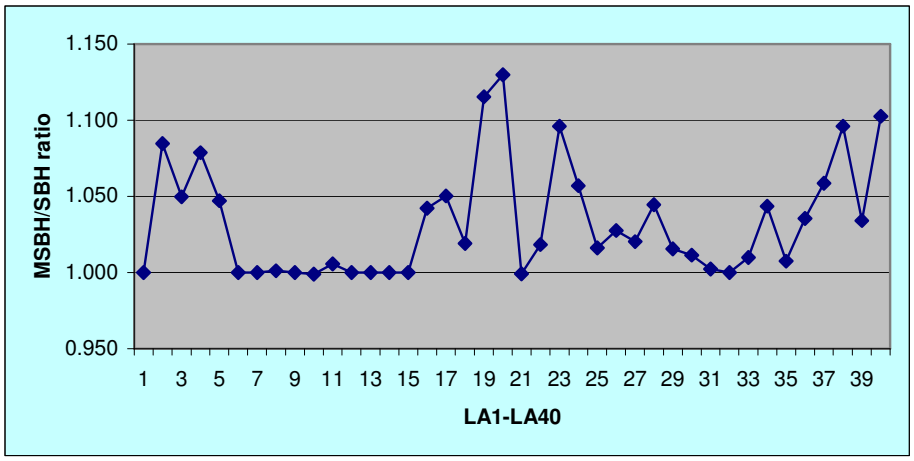


Figure 6.1 Quality of the solutions obtained by the MSBH

The computation times vary from 0.52 seconds to 76.7 seconds for the original SBH, from 9 seconds to 66 seconds for the MSBH. For example, the MSBH requires 46 seconds to schedule a 15 machines, 15 jobs instance (LA40), whereas the original SBH requires 76.7 seconds. We compared the computational time results using the

paired t-test. The difference of the MSBH solution times from those of the SBH is not statistically significant at the 1% level of significance, whereas solution qualities are statistically significant. Because it has a sequencing based algorithm, the minimum time is 9 seconds for small size problems. With increase in the number of operations, the MSBH requires less computation times. Furthermore, for the MSBH and the original SBH, because the programming languages and computer features used to solve the instances are different, it is not robust to compare the solution times.

The computation times for the SBH methods increase rapidly with the number of jobs and number of machines. For example, it rises up to an average of more than 4000 seconds (1.1 hours) on the 50 job problems for the modified SBH of Balas et al. (1995), who has inserted delayed precedence constraints to the SBH of Adams et al. (1988). (Demirkol et al., 1997).

For even larger real world problems with hundreds of jobs, the amount of time required to obtain a good solution may prove unacceptable by the SBH. Therefore, additional research needs to be done to investigate methods for reducing the SBH' s solution time while maintaining solution quality for even larger problem instances. With the increase in the number of operations, the MSBH could find the results in less computation times than the SBH. For example, for 50 jobs - 20 machines instances, the computational times for the MSBH is about 2000 seconds for the RJSSPs, as it given in Section 6.3.

6.3 Comparison Of The Proposed Methods For The Reentrant Job Shop Scheduling Problem For Small Size Test Problems

In order to compare the results of the proposed methods with optimal solutions for the RJSSP, a set of small size test problems are generated. In these problems, the number of jobs differs from 3 to 5, and the number of machines differs from 3 to 10. The optimal results are obtained by the branch and bound algorithm of LINGO 8.0 software package using the mixed integer programming (MIP) formulation of Pan &

Chen (2005), which is described in Section 3.1. Furthermore, these problems are solved by the MSBH, the MSBH with Total Machine Load (MSBH_TML), the MSBH with Total Infeasibility (MSBH_TOTINF) and the MRHP. The results are compared with each other and with optimal results to see how far the modified methods are from the optimum.

As we described in the previous sections, an important component of the MSBH is the resequencing step. The resequencing step is clearly computationally intensive, involving many single machine problems and longest path calculations. We limit the number of resequencing cycles to six as described in Section 4.1. In the proposed methods, full reoptimization with maximum six rescheduling cycles is applied.

Furthermore, to evaluate the importance of the resequencing step to the performance of the MSBH, we consider three different resequencing schemes. We first consider the MSBH with full resequencing (MSBH(FR)). Secondly, the MSBH is applied with no resequencing procedure (MSBH(NR)). In addition to no and full resequencing, different number of cycles from one to five are tested. The results are compared with each other to see the effect of using different resequencing cycles on solution quality and computational time.

In order to evaluate the solution and time quality of the MRHP, the method is tested for different number of time windows (two and three time windows) and for the rules for assignment of operations to time windows (Section 5.1). These rules are as follows:

- Time window with a fixed number of operations per job (MRHP_FJOB).
- Time window with a balance number of operations per machine (MRHP_BMCH).
- Time window with a balanced workload for each job (MRHP_BJOB).

The numerical values of problem parameters are generated according to the following scheme. Processing times are randomly generated from a discrete uniform

distribution over [1,20]; the technological constraints for each job are assigned in a randomly determined order. A problem is denoted by n x m x [m+1; max_N], where the bracket represents the range on the number of operations for each job. The number of jobs is denoted by n, and the number of machines is denoted by m. The number of operations of a job is set to be at least m+1 to ensure reentry, and the value max_N is a user input parameter.

The undermentioned sizes that are given in Pan & Chen (2005) are generated and solved.

- 3 x 5 x [6,30]
- 3 x 5 x [6,40]
- 3 x 10 x [11,20]
- 3 x 10 x [11,30]
- 5 x 3 x [4,6]
- 5 x 3 x [4,9]
- 5 x 5 x [6,8]
- 5 x 5 x [6,10]

The experiment consists of 8 different problem sizes which are given above. For each problem size, 20 test problems are generated. Thus, the model is applied to total 160 randomly generated problem instances.

6.3.1 Details Of The Implementation And The Computational Results

All the results of these experiments are given in Appendix B. Solution quality and time for each instance are expressed as the ratio of C_{max} and CPU time obtained by the proposed algorithms to optimal ones obtained by the MIP model using LINGO. The relative error RE(%) was calculated for each instance of problem, as a percentage by which the solution obtained is above the optimum value. For example, for 3x5x[6,30] instances the comparison of the MSBH and optimal results are given in Table 6.4. As a summary, Figures 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8 and 6.9 show the solution ratios of the 8 job-machine sets successively.

Table 6.4 MSBH – MIP comparison of the 3x5x[6,30] instances

	MIP		MSBH		MSBH-MIP comparison			
					C_{max}		time	
	C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)
1	294	856	327	13	1.112	11.2	0.015	-98.5
2	333	108	349	12	1.048	4.8	0.111	-88.9
3	292	282	317	13	1.086	8.6	0.046	-95.4
4	333	81	352	12	1.057	5.7	0.148	-85.2
5	311	159	341	13	1.096	9.6	0.082	-91.8
6	297	72	328	12	1.104	10.4	0.167	-83.3
7	277	90	291	12	1.051	5.1	0.133	-86.7
8	302	81	323	12	1.070	7.0	0.148	-85.2
9	294	306	331	13	1.126	12.6	0.042	-95.8
10	139	90	153	11	1.101	10.1	0.122	-87.8
11	124	83	127	10	1.024	2.4	0.120	-88.0
12	370	1297	393	13	1.062	6.2	0.010	-99.0
13	305	66	321	12	1.052	5.2	0.182	-81.8
14	312	902	336	15	1.077	7.7	0.017	-98.3
15	265	52	267	11	1.008	0.8	0.212	-78.8
16	211	91	216	11	1.024	2.4	0.121	-87.9
17	196	81	210	11	1.071	7.1	0.136	-86.4
18	324	122	339	13	1.046	4.6	0.107	-89.3
19	338	301	362	14	1.071	7.1	0.047	-95.3
20	228	58	242	12	1.061	6.1	0.207	-79.3

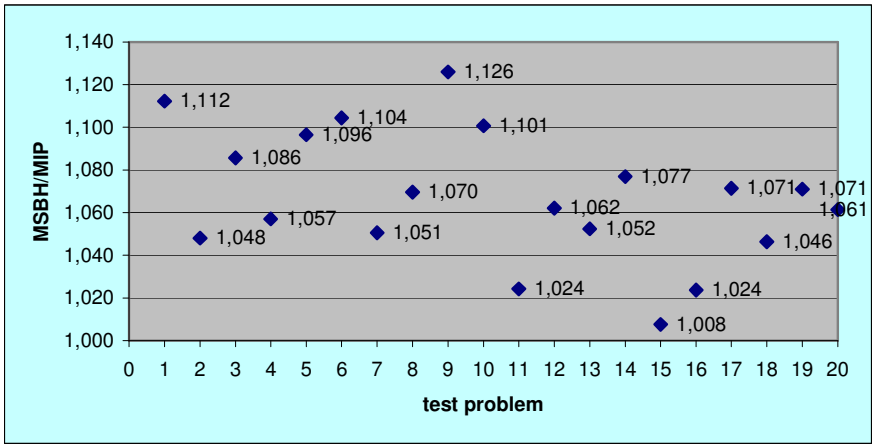


Figure 6.2 The solution ratios for the 3x5x[6,30] instances

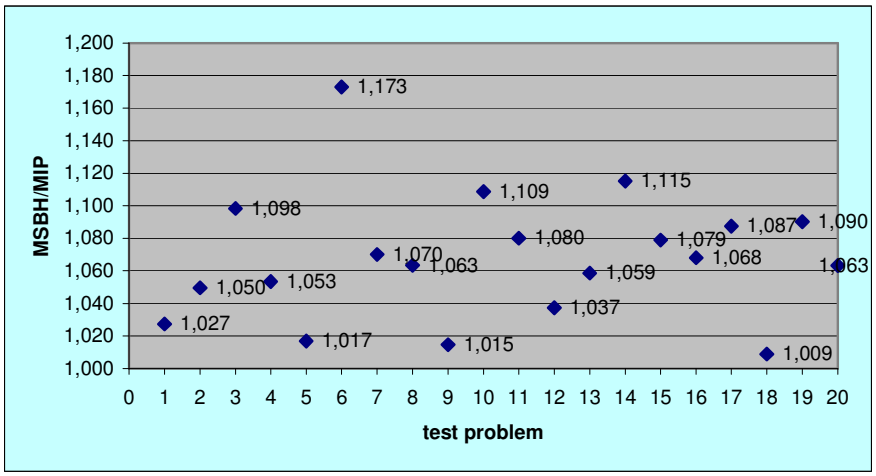


Figure 6.3 The solution ratios for the 3x5x[6,40] instances

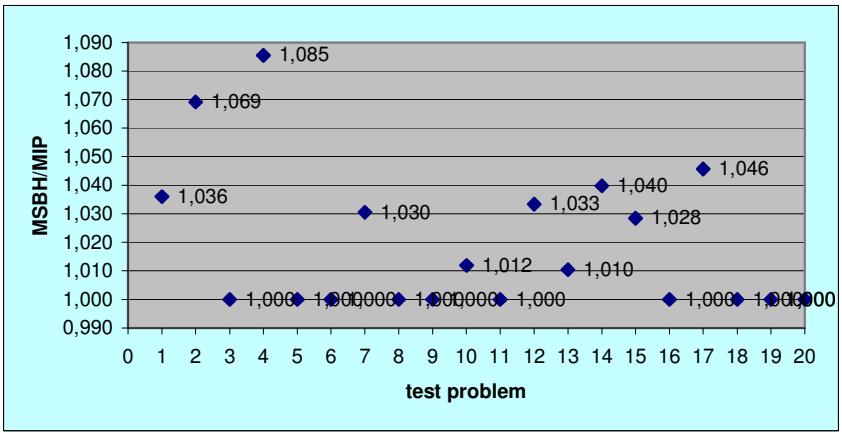


Figure 6.4. The solution ratios for the 3x10x[11,20] instances

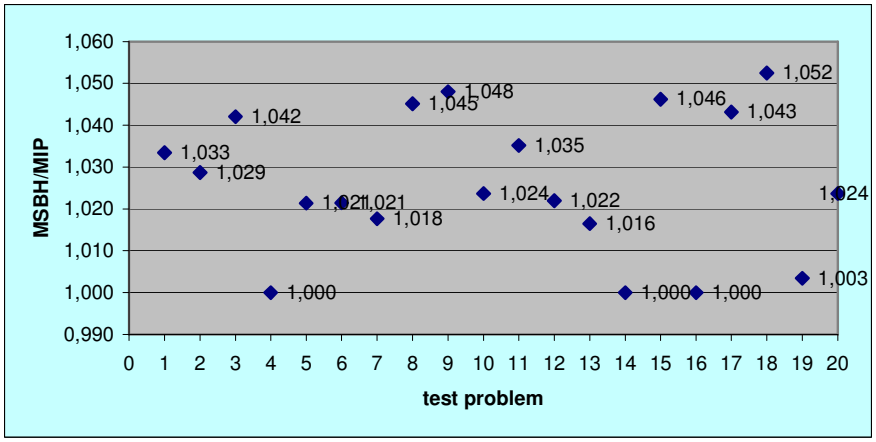


Figure 6.5. The solution ratios for the 3x10x[11,30] instances

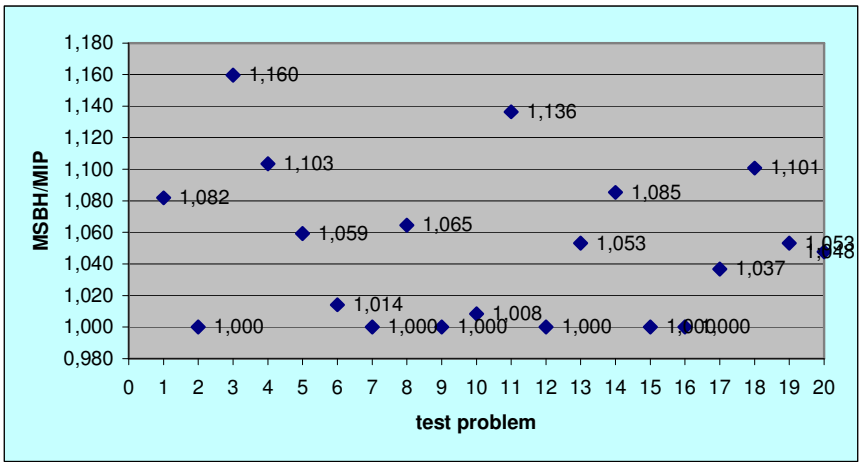


Figure 6.6 The solution ratios for the 5x3x[4,6] instances

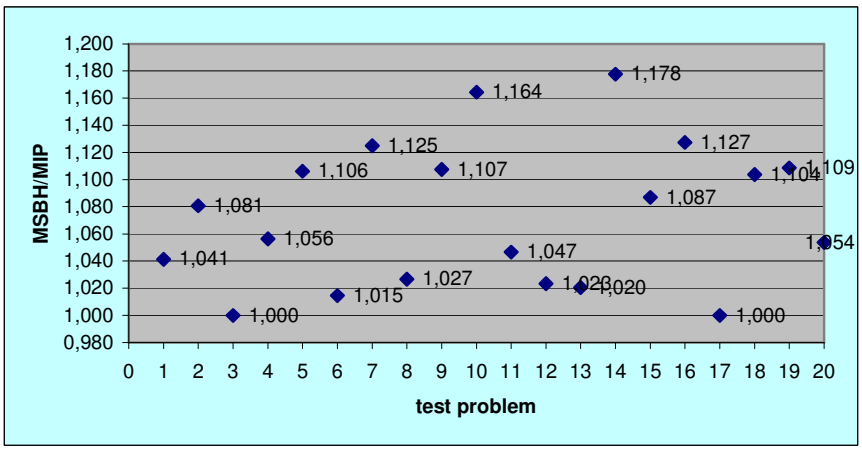


Figure 6.7 The solution ratios for the 5x3x[4,9] instances

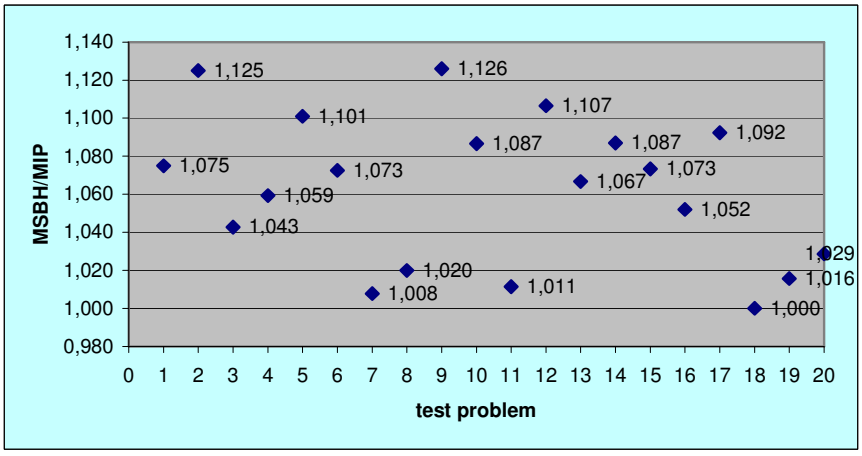


Figure 6.8 The solution ratios for the 5x5x[6,8] instances

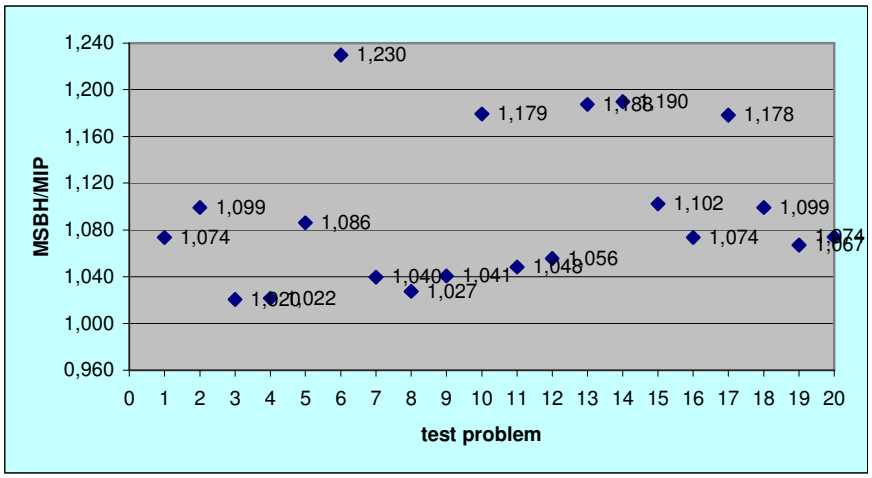


Figure 6.9 The solution ratios for the 5x5x[6,10] instances

20 problems are solved for each job-machine combination and the average, minimum and maximum values for the C_{max} and solution time ratios and relative errors are displayed in the following tables. Tables 6.5 and 6.6 give the comparisons of the results for the 3x5x[6,30] instances. From Table 6.5, for the MSBH, we see that the average solution gap from optimum is 6.74%. The average decrease in the solution times is 89.14%. As a result, when we go away from the solution quality 6.74% in average, we obtain 89.14% improvement on the solution time. The RE(%) from optimum for the MSBH_TOTINF (6.55%) is less than the MSBH_TML (6.87%). But solution times obtained by the MSBH_TML are shorter than the MSBH_TOTINF. As it is seen in Table 6.6, among the MRHP methods, the MRHP_FJOB with two time windows gives the best solution with 10.62% gap when compared with optimum. The largest gap belongs to the MRHP_BJOB with three time windows, which is 17.86%. The quality of the solutions obtained by three time windows are worse than those obtained by two time windows, but the solution times are shorter.

Table 6.5 Comparison of the MSBH methods for the 3x5x[6,30] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MSBH - MIP	**avg.**	1.067	6.74	0.109	-89.14
	min.	1.008	0.75	0.010	-99.00
	max.	1.126	12.59	0.212	-78.85
MSBH_TML - MIP	**avg.**	1.069	6.87	0.103	-89.69
	min.	1.008	0.75	0.010	-99.00
	max.	1.148	14.79	0.212	-78.85
MSBH_TOTINF - MIP	**avg.**	1.066	6.55	0.127	-87.25
	min.	1.019	1.92	0.014	-98.61
	max.	1.125	12.54	0.250	-75.00
MSBH_TML - MSBH	**avg.**	1.001	0.13	0.942	-5.82
	min.	0.957	-4.29	0.846	-15.38
	max.	1.047	4.69	1.000	0.00
MSBH_TOTINF - MSBH	**avg.**	0.999	-0.14	1.198	19.83
	min.	0.945	-5.50	1.083	8.33
	max.	1.059	5.85	1.385	38.46
MSBH_TOTINF – MSBH_TML	**avg.**	0.998	-0.24	1.277	27.71
	min.	0.931	-6.93	1.083	8.33
	max.	1.048	4.79	1.455	45.45

Table 6.6 Comparison of the MRHP methods for the 3x5x[6,30] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
*MRHP_FJOB (2window) - MIP	**avg.**	1.106	10.62	0.233	-76.69
	min.	1.000	0.00	0.022	-97.84
	max.	1.259	25.85	0.466	-53.45
MRHP_FJOB(3window) - MIP	**avg.**	1.164	16.41	0.218	-78.18
	min.	1.008	0.75	0.019	-98.07
	max.	1.353	35.25	0.462	-53.85
**MRHP_BMCH (2window) - MIP	**avg.**	1.125	12.54	0.232	-76.82
	min.	1.000	0.00	0.019	-98.07
	max.	1.330	33.00	0.466	-53.45
MRHP_BMCH(3window) - MIP	**avg.**	1.149	14.90	0.227	-77.35
	min.	1.000	0.00	0.020	-98.00
	max.	1.331	33.14	0.442	-55.77
***MRHP_BJOB (2window) - MIP	**avg.**	1.142	14.19	0.226	-77.41
	min.	1.000	0.00	0.020	-98.00
	max.	1.354	35.35	0.442	-55.77
MRHP_BJOB(3window) - MIP	**avg.**	1.179	17.86	0.215	-78.52
	min.	1.043	4.33	0.020	-98.00
	max.	1.320	31.99	0.442	-55.77
MRHP_FJOB(2window) - MSBH	**avg.**	1.036	3.59	2.116	111.59
	min.	0.942	-5.79	1.857	85.71
	max.	1.153	15.28	2.500	150.00
MRHP_FJOB(3window) - MSBH	**avg.**	1.090	9.01	1.955	95.46
	min.	0.946	-5.36	1.692	69.23
	max.	1.233	23.34	2.200	120.00
MRHP_BMCH(2window) - MSBH	**avg.**	1.054	5.37	2.087	108.69
	min.	0.942	-5.79	1.846	84.62
	max.	1.204	20.43	2.333	133.33
MRHP_BMCH(3window) - MSBH	**avg.**	1.076	7.61	2.036	103.59
	min.	0.942	-5.79	1.714	71.43
	max.	1.243	24.31	2.500	150.00
MRHP_BJOB(2window) - MSBH	**avg.**	1.069	6.95	0.226	-77.41
	min.	0.955	-4.46	0.020	-98.00
	max.	1.315	31.50	0.442	-55.77
MRHP_BJOB(3window) - MSBH	**avg.**	1.104	10.40	1.948	94.78
	min.	0.993	-0.69	1.714	71.43
	max.	1.276	27.56	2.300	130.00
MRHP_FJOB(3window) - MRHP_FJOB(2window)	**avg.**	1.054	5.36	0.926	-7.44
	min.	0.960	-4.02	0.786	-21.43
	max.	1.196	19.57	1.043	4.35
MRHP_BMCH(3window) - MRHP_BMCH(2window)	**avg.**	1.023	2.31	0.976	-2.40
	min.	0.911	-8.93	0.893	-10.71
	max.	1.240	23.97	1.087	8.70
MRHP_BJOB(3window) - MRHP_BJOB(2window)	**avg.**	1.035	3.50	0.949	-5.13
	min.	0.929	-7.07	0.828	-17.24
	max.	1.156	15.60	1.100	10.00

* Time window with a fixed number of operations per job.
** Time window with a balanced number of operations per machine.
*** Time window with a balanced workload for each job.

Tables 6.7 and 6.8 give the comparisons of the results for the 3x5x[6,40] instances. The average solution gap is 6.81% for both the MSBH and the MSBH_TOTINF, whereas the solution gap of the MSBH_TML is 7.10%. Furthermore the solution times obtained with the MSBH are shorter than the MSBH_TML and the MSBH_TOTINF. Among the MRHP methods, the MRHP_BJOB with two time windows gives the best solution quality from optimum with 11.40% solution gap, whereas the worst one belongs to the MRHP_FJOB with three time windows (16.27%). The solution time differences between two and three time windows are not significant, but the solution times for three time windows are a little bit shorter than for two time windows.

Table 6.7 Comparison of the MSBH methods for the 3x5x[6,40] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MSBH - MIP	**avg.**	1.068	6.81	0.047	-95.34
	min.	1.009	0.88	0.001	-99.91
	max.	1.173	17.30	0.119	-88.07
MSBH_TML - MIP	**avg.**	1.071	7.10	0.048	-95.20
	min.	1.007	0.72	0.001	-99.91
	max.	1.148	14.81	0.121	-87.88
MSBH_TOTINF - MIP	**avg.**	1.068	6.81	0.064	-93.64
	min.	1.009	0.88	0.001	-99.87
	max.	1.129	12.95	0.172	-82.83
MSBH_TML - MSBH	**avg.**	1.003	0.29	1.035	3.46
	min.	0.953	-4.71	0.923	-7.69
	max.	1.053	5.32	1.154	15.38
MSBH_TOTINF - MSBH	**avg.**	1.000	0.02	1.396	39.64
	min.	0.957	-4.29	1.231	23.08
	max.	1.041	4.05	1.750	75.00
MSBH_TOTINF – MSBH_TML	**avg.**	0.998	-0.23	1.351	35.09
	min.	0.962	-3.83	1.167	16.67
	max.	1.067	6.71	1.615	61.54

Table 6.8 Comparison of the MRHP methods for the 3x5x[6,40] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MRHP_FJOB(2window) - MIP	**avg.**	1.123	12.28	0.103	-89.72
	min.	1.024	2.44	0.002	-99.82
	max.	1.305	30.50	0.253	-74.75
MRHP_FJOB(3window) - MIP	**avg.**	1.163	16.27	0.102	-89.80
	min.	1.047	4.71	0.002	-99.83
	max.	1.386	38.58	0.263	-73.74
MRHP_BMCH(2window) - MIP	**avg.**	1.123	12.30	0.098	-90.17
	min.	1.022	2.17	0.002	-99.84
	max.	1.296	29.56	0.242	-75.76
MRHP_BMCH(3window) - MIP	**avg.**	1.161	16.08	0.096	-90.36
	min.	1.025	2.48	0.002	-99.84
	max.	1.421	42.14	0.242	-75.76
MRHP_BJOB(2window) - MIP	**avg.**	1.114	11.40	0.100	-90.00
	min.	1.009	0.88	0.002	-99.82
	max.	1.308	30.85	0.263	-73.74
MRHP_BJOB(3window) - MIP	**avg.**	1.122	12.20	0.099	-90.13
	min.	1.010	0.97	0.002	-99.82
	max.	1.255	25.47	0.263	-73.74
MRHP_FJOB(2window) - MSBH	**avg.**	1.051	5.13	2.177	117.75
	min.	0.983	-1.67	1.933	93.33
	max.	1.170	17.02	2.455	145.45
MRHP_FJOB(3window) - MSBH	**avg.**	1.090	8.99	2.123	112.31
	min.	0.944	-5.56	1.800	80.00
	max.	1.309	30.90	2.455	145.45
MRHP_BMCH(2window) - MSBH	**avg.**	1.051	5.11	2.072	107.19
	min.	0.969	-3.06	1.733	73.33
	max.	1.140	13.99	2.417	141.67
MRHP_BMCH(3window) - MSBH	**avg.**	1.088	8.76	2.025	102.49
	min.	0.925	-7.52	1.733	73.33
	max.	1.276	27.55	2.333	133.33
MRHP_BJOB(2window) - MSBH	**avg.**	1.043	4.30	2.110	111.04
	min.	0.984	-1.63	1.867	86.67
	max.	1.203	20.33	2.545	154.55
MRHP_BJOB(3window) - MSBH	**avg.**	1.051	5.07	2.080	107.99
	min.	0.962	-3.82	1.833	83.33
	max.	1.197	19.71	2.364	136.36
MRHP_FJOB(3window) - MRHP_FJOB(2window)	**avg.**	1.037	3.66	0.975	-2.47
	min.	0.945	-5.45	0.920	-8.00
	max.	1.133	13.33	1.042	4.17
MRHP_BMCH(3window) - MRHP_BMCH(2window)	**avg.**	1.035	3.49	0.979	-2.06
	min.	0.873	-12.65	0.828	-17.24
	max.	1.195	19.52	1.077	7.69
MRHP_BJOB(3window) - MRHP_BJOB(2window)	**avg.**	1.010	0.97	0.987	-1.34
	min.	0.839	-16.11	0.917	-8.33
	max.	1.142	14.22	1.042	4.17

Tables 6.9 and 6.10 give the comparisons of the results for the 3x10x[11,20] instances. In average, the MSBH_TML gives the best solution quality with 1.58% solution gap. The average gap for the MSBH is 1.95%, whereas it is 1.82% for the MSBH_TOTINF. Furthermore, the computational times obtained by TML are shorter than the others. Among the MRHP methods, the MRHP_BJOB with two time windows gives the best quality in average (4.15%), whereas a worst case solution gap from optimum is 15.52% obtained by the MRHP_BMCH with three time windows.

Table 6.9 Comparison of the MSBH methods for the 3x10x[11,20] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MSBH - MIP	**avg.**	1.020	1.95	0.614	-38.585
	min.	1.000	0.00	0.317	-68.293
	max.	1.085	8.54	0.923	-7.692
MSBH_TML - MIP	**avg.**	1.016	1.58	0.595	-40.55
	min.	1.000	0.00	0.293	-70.73
	max.	1.060	6.03	1.111	11.11
MSBH_TOTINF - MIP	**avg.**	1.018	1.82	0.996	-0.37
	min.	1.000	0.00	0.488	-51.22
	max.	1.060	6.03	1.889	88.89
MSBH_TML - MSBH	**avg.**	0.997	-0.35	0.907	-9.33
	min.	0.974	-2.55	0.833	-16.67
	max.	1.025	2.53	1.000	0.00
MSBH_TOTINF - MSBH	**avg.**	0.999	-0.10	1.518	51.76
	min.	0.935	-6.47	1.385	38.46
	max.	1.047	4.71	1.833	83.33
MSBH_TOTINF – MSBH_TML	**avg.**	1.002	0.24	1.676	67.60
	min.	0.959	-4.08	1.385	38.46
	max.	1.047	4.71	2.000	100.00

Table 6.10 Comparison of the MRHP methods for the 3x10x[11,20] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MRHP_FJOB(2window) - MIP	**avg.**	1.047	4.69	1.409	40.94
	min.	1.000	0.00	0.610	-39.02
	max.	1.303	30.29	2.889	188.89
MRHP_FJOB(3window) - MIP	**avg.**	1.078	7.75	1.357	35.73
	min.	1.000	0.00	0.585	-41.46
	max.	1.297	29.69	2.889	188.89
MRHP_BMCH(2window) - MIP	**avg.**	1.063	6.25	1.415	41.48
	min.	1.000	0.00	0.634	-36.59
	max.	1.356	35.63	2.700	170.00
MRHP_BMCH(3window) - MIP	**avg.**	1.155	15.52	1.353	35.25
	min.	1.000	0.00	0.610	-39.02
	max.	1.341	34.07	2.778	177.78
MRHP_BJOB(2window) - MIP	**avg.**	1.042	4.15	1.419	41.88
	min.	1.000	0.00	0.659	-34.15
	max.	1.253	25.31	2.889	188.89
MRHP_BJOB(3window) - MIP	**avg.**	1.051	5.14	1.324	32.40
	min.	1.000	0.00	0.634	-36.59
	max.	1.161	16.15	2.556	155.56
MRHP_FJOB(2window) - MSBH	**avg.**	1.027	2.67	2.126	112.56
	min.	0.935	-6.47	1.923	92.31
	max.	1.246	24.59	2.615	161.54
MRHP_FJOB(3window) - MSBH	**avg.**	1.056	5.65	2.038	103.78
	min.	0.974	-2.55	1.833	83.33
	max.	1.284	28.35	2.250	125.00
MRHP_BMCH(2window) - MSBH	**avg.**	1.043	4.31	2.134	113.37
	min.	0.949	-5.09	1.923	92.31
	max.	1.356	35.63	2.417	141.67
MRHP_BMCH(3window) - MSBH	**avg.**	1.133	13.31	2.042	104.17
	min.	0.935	-6.47	1.846	84.62
	max.	1.320	31.96	2.385	138.46
MRHP_BJOB(2window) - MSBH	**avg.**	1.022	2.20	2.130	113.01
	min.	0.954	-4.63	1.923	92.31
	max.	1.253	25.31	2.692	169.23
MRHP_BJOB(3window) - MSBH	**avg.**	1.031	3.13	2.002	100.19
	min.	0.995	-0.50	1.846	84.62
	max.	1.149	14.95	2.167	116.67
MRHP_FJOB(3window) - MRHP_FJOB(2window)	**avg.**	1.031	3.11	0.961	-3.89
	min.	0.882	-11.84	0.824	-17.65
	max.	1.239	23.88	1.042	4.17
MRHP_BMCH(3window) - MRHP_BMCH(2window)	**avg.**	1.091	9.08	0.959	-4.14
	min.	0.940	-6.00	0.862	-13.79
	max.	1.323	32.31	1.080	8.00
MRHP_BJOB(3window) - MRHP_BJOB(2window)	**avg.**	1.012	1.18	0.943	-5.66
	min.	0.847	-15.27	0.743	-25.71
	max.	1.109	10.95	1.040	4.00

Tables 6.11 and 6.12 give the comparisons of the results for the 3x10x[11,30] instances. Among the methods, the best quality obtained by the MSBH_TOTINF with 1.48% solution gap from optimum. The MSBH_TML is the second with 1.88%, and MSBH is the third with 2.62%. As it is seen, the MSBH is outperformed by the MSBH_TML on both solution quality and time, and by the MSBH_TOTINF on solution quality. Among the MRHP methods, the MRHP_BJOB with two time windows outperforms the other rules. The worst quality from optimum is obtained by MRHP_BMCH with three time windows (14.25%).

Table 6.11 Comparison of the MSBH methods for the 3x10x[11,30] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MSBH - MIP	**avg.**	1.026	2.62	0.273	-72.65
	min.	1.000	0.00	0.152	-84.76
	max.	1.052	5.24	0.333	-66.67
MSBH_TML - MIP	**avg.**	1.019	1.88	0.219	-78.05
	min.	1.000	0.00	0.143	-85.71
	max.	1.092	9.21	0.280	-72.00
MSBH_TOTINF - MIP	**avg.**	1.015	1.48	0.492	-50.78
	min.	1.000	0.00	0.308	-69.23
	max.	1.046	4.62	0.640	-36.00
MSBH_TML - MSBH	**avg.**	0.993	-0.70	0.812	-18.76
	min.	0.950	-4.98	0.688	-31.25
	max.	1.069	6.87	1.188	18.75
MSBH_TOTINF - MSBH	**avg.**	0.989	-1.09	1.824	82.45
	min.	0.954	-4.58	1.333	33.33
	max.	1.020	1.98	3.063	206.25
MSBH_TOTINF – MSBH_TML	**avg.**	0.996	-0.36	2.246	124.63
	min.	0.936	-6.43	1.750	75.00
	max.	1.045	4.55	3.364	236.36

Table 6.12 Comparison of the MRHP methods for the 3x10x[11,30] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MRHP_FJOB(2window) - MIP	**avg.**	1.072	7.18	0.512	-48.83
	min.	1.000	0.00	0.314	-68.57
	max.	1.220	22.03	0.745	-25.53
MRHP_FJOB(3window) - MIP	**avg.**	1.105	10.53	0.500	-49.96
	min.	1.003	0.34	0.286	-71.43
	max.	1.307	30.71	0.638	-36.17
MRHP_BMCH(2window) - MIP	**avg.**	1.078	7.76	0.521	-47.86
	min.	1.000	0.00	0.324	-67.62
	max.	1.291	29.12	0.683	-31.67
MRHP_BMCH(3window) - MIP	**avg.**	1.142	14.25	0.489	-51.05
	min.	1.000	0.00	0.286	-71.43
	max.	1.443	44.26	0.609	-39.13
MRHP_BJOB(2window) - MIP	**avg.**	1.044	4.38	0.541	-45.90
	min.	1.000	0.00	0.295	-70.48
	max.	1.204	20.36	0.739	-26.09
MRHP_BJOB(3window) - MIP	**avg.**	1.063	6.30	0.506	-49.42
	min.	1.000	0.00	0.276	-72.38
	max.	1.236	23.59	0.680	-32.00
MRHP_FJOB(2window) - MSBH	**avg.**	1.045	4.52	1.874	87.38
	min.	0.954	-4.58	1.733	73.33
	max.	1.212	21.16	2.333	133.33
MRHP_FJOB(3window) - MSBH	**avg.**	1.078	7.76	1.831	83.08
	min.	0.979	-2.08	1.600	60.00
	max.	1.307	30.71	2.000	100.00
MRHP_BMCH(2window) - MSBH	**avg.**	1.051	5.05	1.920	92.03
	min.	0.954	-4.58	1.625	62.50
	max.	1.278	27.80	2.600	160.00
MRHP_BMCH(3window) - MSBH	**avg.**	1.114	11.36	1.791	79.13
	min.	0.988	-1.18	1.625	62.50
	max.	1.409	40.92	1.933	93.33
MRHP_BJOB(2window) - MSBH	**avg.**	1.017	1.74	1.973	97.26
	min.	0.950	-4.98	1.625	62.50
	max.	1.165	16.47	2.267	126.67
MRHP_BJOB(3window) - MSBH	**avg.**	1.036	3.62	1.843	84.32
	min.	0.966	-3.40	1.563	56.25
	max.	1.236	23.59	2.267	126.67
MRHP_FJOB(3window) - MRHP_FJOB(2window)	**avg.**	1.033	3.32	0.979	-2.07
	min.	0.834	-16.61	0.857	-14.29
	max.	1.178	17.81	1.111	11.11
MRHP_BMCH(3window) - MRHP_BMCH(2window)	**avg.**	1.064	6.35	0.946	-5.45
	min.	0.869	-13.11	0.683	-31.71
	max.	1.306	30.61	1.077	7.69
MRHP_BJOB(3window) - MRHP_BJOB(2window)	**avg.**	1.020	1.99	0.935	-6.52
	min.	0.859	-14.14	0.852	-14.81
	max.	1.119	11.89	1.063	6.25

Tables 6.13 and 6.14 give the comparisons of the results for the 5x3x[4,6] instances. The MSBH, which has a 5.02% solution gap from optimum, outperforms the other methods. The MSBH_TML outperforms all the methods with 0.136 ratio for CPU times. The worst case for quality is obtained by the MRHP_BJOB with three time windows which has 20.31% average gap from optimum.

Table 6.13 Comparison of the MSBH methods for the 5x3x[4,6] instances

		C_{max}		time	
		C_{max}	**RE(%)**	**ratio**	**RE(%)**
MSBH - MIP	**avg.**	1.050	5.02	0.158	-84.17
	min.	1.000	0.00	0.007	-99.28
	max.	1.160	15.96	0.550	-45.00
MSBH_TML - MIP	**avg.**	1.059	5.86	0.136	-86.45
	min.	1.000	0.00	0.006	-99.41
	max.	1.160	15.96	0.476	-52.38
MSBH_TOTINF - MIP	**avg.**	1.058	5.77	0.160	-83.99
	min.	1.000	0.00	0.007	-99.28
	max.	1.197	19.67	0.550	-45.00
MSBH_TML - MSBH	**avg.**	1.008	0.84	0.852	-14.84
	min.	0.986	-1.41	0.750	-25.00
	max.	1.076	7.63	0.909	-9.09
MSBH_TOTINF - MSBH	**avg.**	1.008	0.78	1.012	1.20
	min.	0.915	-8.45	0.917	-8.33
	max.	1.106	10.61	1.100	10.00
MSBH_TOTINF – MSBH_TML	**avg.**	1.000	-0.04	1.191	19.06
	min.	0.929	-7.14	1.100	10.00
	max.	1.098	9.77	1.222	22.22

Table 6.14 Comparison of the MRHP methods for the 5x3x[4,6] instances

		C_{max}		time	
		ratio	**RE(%)**	ratio	RE(%)
MRHP_FJOB(2window) - MIP	**avg.**	1.107	10.75	0.338	-66.21
	min.	1.000	0.00	0.015	-98.49
	max.	1.273	27.27	1.143	14.29
MRHP_FJOB(3window) - MIP	**avg.**	1.132	13.20	0.323	-67.66
	min.	1.000	0.00	0.015	-98.46
	max.	1.386	38.64	1.100	10.00
MRHP_BMCH(2window) - MIP	**avg.**	1.134	13.42	0.332	-66.80
	min.	1.000	0.00	0.016	-98.42
	max.	1.409	40.91	1.143	14.29
MRHP_BMCH(3window) - MIP	**avg.**	1.138	13.79	0.278	-72.23
	min.	1.000	0.00	0.000	-100.00
	max.	1.375	37.50	1.100	10.00
MRHP_BJOB(2window) - MIP	**avg.**	1.143	14.27	0.352	-64.77
	min.	1.029	2.88	0.016	-98.42
	max.	1.254	25.42	1.100	10.00
MRHP_BJOB(3window) - MIP	**avg.**	1.203	20.31	0.324	-67.65
	min.	1.000	0.00	0.014	-98.55
	max.	1.458	45.76	1.100	10.00
MRHP_FJOB(2window) - MSBH	**avg.**	1.055	5.48	2.132	113.23
	min.	0.949	-5.05	1.909	90.91
	max.	1.161	16.10	2.400	140.00
MRHP_FJOB(3window) - MSBH	**avg.**	1.078	7.83	2.044	104.42
	min.	0.936	-6.42	1.917	91.67
	max.	1.220	22.00	2.200	120.00
MRHP_BMCH(2window) - MSBH	**avg.**	1.079	7.91	2.144	114.36
	min.	1.000	0.00	1.833	83.33
	max.	1.240	24.00	2.400	140.00
MRHP_BMCH(3window) - MSBH	**avg.**	1.083	8.32	2.044	104.42
	min.	0.972	-2.80	1.917	91.67
	max.	1.210	21.00	2.200	120.00
MRHP_BJOB(2window) - MSBH	**avg.**	1.090	8.99	2.275	127.47
	min.	0.990	-1.01	2.000	100.00
	max.	1.254	25.42	2.455	145.45
MRHP_BJOB(3window) - MSBH	**avg.**	1.147	14.72	2.092	109.17
	min.	0.990	-1.01	1.833	83.33
	max.	1.458	45.76	2.400	140.00
MRHP_FJOB(3window) - MRHP_FJOB(2window)	**avg.**	1.023	2.31	0.959	-4.10
	min.	0.858	-14.17	0.840	-16.00
	max.	1.124	12.36	1.087	8.70
MRHP_BMCH(3window) - MRHP_BMCH(2window)	**avg.**	1.006	0.56	0.908	-9.16
	min.	0.866	-13.39	0.000	-100.00
	max.	1.124	12.36	1.136	13.64
MRHP_BJOB(3window) - MRHP_BJOB(2window)	**avg.**	1.058	5.75	0.922	-7.76
	min.	0.813	-18.70	0.759	-24.14
	max.	1.293	29.32	1.043	4.35

Tables 6.15 and 6.16 give the comparisons of the results for the 5x3x[4,9] instances. The MSBH_TML outperforms all the methods with 6.18% average gap from optimum. The solution quality gap of the MSBH is 7.35%. The worst case is obtained by the MRHP_FJOB with three time windows (19.54%) on solution quality.

Table 6.15 Comparison of the MSBH methods for the 5x3x[4,9] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MSBH - MIP	**avg.**	1.074	7.35	0.011	-98.87
	min.	1.000	0.00	0.000	-99.96
	max.	1.178	17.76	0.092	-90.76
MSBH_TML - MIP	**avg.**	1.062	6.18	0.011	-98.94
	min.	1.000	0.00	0.000	-99.96
	max.	1.178	17.76	0.084	-91.60
MSBH_TOTINF - MIP	**avg.**	1.076	7.56	0.011	-98.86
	min.	1.000	0.00	0.000	-99.96
	max.	1.178	17.76	0.092	-90.76
MSBH_TML - MSBH	**avg.**	0.989	-1.05	0.938	-6.25
	min.	0.904	-9.60	0.833	-16.67
	max.	1.022	2.24	1.000	0.00
MSBH_TOTINF - MSBH	**avg.**	1.002	0.21	1.010	0.98
	min.	0.904	-9.60	0.917	-8.33
	max.	1.073	7.28	1.091	9.09
MSBH_TOTINF – MSBH_TML	**avg.**	1.014	1.35	1.080	7.95
	min.	0.962	-3.77	1.000	0.00
	max.	1.142	14.15	1.200	20.00

Table 6.16 Comparison of the MRHP methods for the 5x3x[4,9] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MRHP_FJOB(2window) - MIP	**avg.**	1.160	15.99	0.024	-97.63
	min.	1.019	1.89	0.001	-99.91
	max.	1.479	47.93	0.185	-81.51
MRHP_FJOB(3window) - MIP	**avg.**	1.195	19.54	0.023	-97.66
	min.	1.047	4.72	0.001	-99.91
	max.	1.400	40.00	0.185	-81.51
MRHP_BMCH(2window) - MIP	**avg.**	1.151	15.07	0.024	-97.63
	min.	1.000	0.00	0.001	-99.91
	max.	1.423	42.31	0.185	-81.51
MRHP_BMCH(3window) - MIP	**avg.**	1.193	19.28	0.024	-97.64
	min.	1.047	4.73	0.001	-99.90
	max.	1.529	52.88	0.185	-81.51
MRHP_BJOB(2window) - MIP	**avg.**	1.172	17.22	0.023	-97.66
	min.	1.020	1.99	0.001	-99.91
	max.	1.306	30.58	0.185	-81.51
MRHP_BJOB(3window) - MIP	**avg.**	1.193	19.26	0.023	-97.67
	min.	1.029	2.90	0.001	-99.91
	max.	1.430	42.97	0.185	-81.51
MRHP_FJOB(2window) - MSBH	**avg.**	1.081	8.12	2.082	108.18
	min.	0.904	-9.62	2.000	100.00
	max.	1.336	33.58	2.455	145.45
MRHP_FJOB(3window) - MSBH	**avg.**	1.115	11.53	2.041	104.05
	min.	0.929	-7.11	2.000	100.00
	max.	1.344	34.44	2.273	127.27
MRHP_BMCH(2window) - MSBH	**avg.**	1.073	7.27	2.103	110.34
	min.	0.916	-8.37	2.000	100.00
	max.	1.279	27.92	2.455	145.45
MRHP_BMCH(3window) - MSBH	**avg.**	1.112	11.18	2.077	107.73
	min.	0.975	-2.51	1.917	91.67
	max.	1.359	35.90	2.273	127.27
MRHP_BJOB(2window) - MSBH	**avg.**	1.093	9.29	2.019	101.86
	min.	1.000	0.00	1.818	81.82
	max.	1.233	23.33	2.455	145.45
MRHP_BJOB(3window) - MSBH	**avg.**	1.113	11.32	2.024	102.39
	min.	0.947	-5.33	1.833	83.33
	max.	1.397	39.69	2.182	118.18
MRHP_FJOB(3window) - MRHP_FJOB(2window)	**avg.**	1.035	3.48	0.981	-1.92
	min.	0.821	-17.88	0.828	-17.24
	max.	1.230	22.97	1.045	4.55
MRHP_BMCH(3window) - MRHP_BMCH(2window)	**avg.**	1.040	4.04	0.990	-1.00
	min.	0.884	-11.59	0.828	-17.24
	max.	1.219	21.90	1.091	9.09
MRHP_BJOB(3window) - MRHP_BJOB(2window)	**avg.**	1.020	2.05	1.000	0.04
	min.	0.807	-19.32	0.706	-29.41
	max.	1.351	35.06	1.100	10.00

Tables 6.17 and 6.18 give the comparisons of the results for the 5x5x[6,8] instances. The MSBH outperforms the MSBH_TML and the MSBH_TOTINF on solution quality (6.25%), and it is outperformed by the MSBH_TML and the MSBH_TOTINF on computational times. The worst quality is obtained by the MRHP_BJOB with three time windows, which has 22.39% average solution gap. An interesting situation is, the MRHP_BMCH with three time windows gives better quality in longer computational times than the MRHP_BMCH with two time windows.

Table 6.17 Comparison of the MSBH methods for the 5x5x[6,8] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MSBH - MIP	**avg.**	1.062	6.25	0.220	-78.05
	min.	1.000	0.00	0.060	-93.96
	max.	1.126	12.59	0.857	-14.29
MSBH_TML - MIP	**avg.**	1.076	7.62	0.198	-80.22
	min.	1.000	0.00	0.058	-94.19
	max.	1.242	24.17	0.786	-21.43
MSBH_TOTINF - MIP	**avg.**	1.081	8.12	0.218	-78.24
	min.	1.000	0.00	0.064	-93.60
	max.	1.173	17.31	0.857	-14.29
MSBH_TML - MSBH	**avg.**	1.013	1.33	0.897	-10.26
	min.	0.959	-4.10	0.769	-23.08
	max.	1.155	15.50	1.000	0.00
MSBH_TOTINF - MSBH	**avg.**	1.018	1.79	0.986	-1.42
	min.	0.959	-4.10	0.846	-15.38
	max.	1.085	8.53	1.091	9.09
MSBH_TOTINF – MSBH_TML	**avg.**	1.005	0.54	1.099	9.86
	min.	0.940	-6.04	1.000	0.00
	max.	1.089	8.93	1.200	20.00

Table 6.18 Comparison of the MRHP methods for the 5x5x[6,8] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MRHP_FJOB(2window) - MIP	**avg.**	1.106	10.62	0.434	-56.58
	min.	1.033	3.28	0.121	-87.91
	max.	1.323	32.29	1.571	57.14
MRHP_FJOB(3window) - MIP	**avg.**	1.147	14.73	0.430	-57.00
	min.	1.000	0.00	0.121	-87.91
	max.	1.406	40.63	1.571	57.14
MRHP_BMCH(2window) - MIP	**avg.**	1.165	16.51	0.435	-56.55
	min.	1.048	4.81	0.121	-87.91
	max.	1.375	37.50	1.571	57.14
MRHP_BMCH(3window) - MIP	**avg.**	1.156	15.61	0.432	-56.82
	min.	1.000	0.00	0.121	-87.91
	max.	1.385	38.54	1.500	50.00
MRHP_BJOB(2window) - MIP	**avg.**	1.182	18.17	0.438	-56.24
	min.	1.064	6.42	0.126	-87.36
	max.	1.467	46.67	1.643	64.29
MRHP_BJOB(3window) - MIP	**avg.**	1.224	22.39	0.436	-56.37
	min.	1.078	7.83	0.126	-87.36
	max.	1.462	46.15	1.643	64.29
MRHP_FJOB(2window) - MSBH	**avg.**	1.043	4.30	1.989	98.90
	min.	0.933	-6.67	1.692	69.23
	max.	1.257	25.74	2.182	118.18
MRHP_FJOB(3window) - MSBH	**avg.**	1.081	8.09	1.980	97.95
	min.	0.956	-4.44	1.615	61.54
	max.	1.337	33.66	2.091	109.09
MRHP_BMCH(2window) - MSBH	**avg.**	1.098	9.84	1.996	99.62
	min.	0.963	-3.70	1.769	76.92
	max.	1.307	30.69	2.167	116.67
MRHP_BMCH(3window) - MSBH	**avg.**	1.089	8.91	1.997	99.74
	min.	0.932	-6.77	1.750	75.00
	max.	1.279	27.91	2.091	109.09
MRHP_BJOB(2window) - MSBH	**avg.**	1.113	11.26	0.438	-56.24
	min.	0.972	-2.75	0.126	-87.36
	max.	1.375	37.50	1.643	64.29
MRHP_BJOB(3window) - MSBH	**avg.**	1.153	15.29	1.997	99.73
	min.	0.992	-0.80	1.692	69.23
	max.	1.343	34.31	2.182	118.18
MRHP_FJOB(3window) - MRHP_FJOB(2window)	**avg.**	1.037	3.71	0.996	-0.39
	min.	0.932	-6.84	0.917	-8.33
	max.	1.176	17.59	1.045	4.55
MRHP_BMCH(3window) - MRHP_BMCH(2window)	**avg.**	0.997	-0.30	1.001	0.08
	min.	0.803	-19.66	0.923	-7.69
	max.	1.279	27.88	1.045	4.55
MRHP_BJOB(3window) - MRHP_BJOB(2window)	**avg.**	1.042	4.20	0.998	-0.17
	min.	0.835	-16.48	0.957	-4.35
	max.	1.261	26.09	1.091	9.09

Tables 6.19 and 6.20 give the comparisons of the results for the 5x5x[6,10] instances. The MSBH outperforms the other methods on solution quality (9.47%), whereas the MSBH_TML outperforms all the methods on computational times with average 0.106 ratio. The worst quality is obtained by the MRHP_BJOB with three time windows (22.17%). The MRHP methods with three time windows give worse quality solutions in less computational times when they are compared with the solutions of two time windows.

Table 6.19 Comparison of the MSBH methods for the 5x5x[6,10] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MSBH - MIP	**avg.**	1.095	9.47	0.114	-88.64
	min.	1.020	2.04	0.001	-99.86
	max.	1.230	22.96	0.478	-52.17
MSBH_TML - MIP	**avg.**	1.096	9.61	0.106	-89.39
	min.	1.020	2.04	0.001	-99.88
	max.	1.230	22.96	0.435	-56.52
MSBH_TOTINF - MIP	**avg.**	1.105	10.46	0.114	-88.60
	min.	1.020	2.04	0.001	-99.87
	max.	1.240	24.03	0.522	-47.83
MSBH_TML - MSBH	**avg.**	1.002	0.23	0.921	-7.94
	min.	0.910	-9.04	0.833	-16.67
	max.	1.105	10.48	1.000	0.00
MSBH_TOTINF - MSBH	**avg.**	1.010	1.02	0.985	-1.53
	min.	0.901	-9.87	0.846	-15.38
	max.	1.120	11.97	1.091	9.09
MSBH_TOTINF – MSBH_TML	**avg.**	1.008	0.84	1.072	7.23
	min.	0.907	-9.27	1.000	0.00
	max.	1.120	11.97	1.200	20.00

Table 6.20 Comparison of the MRHP methods for the 5x5x[6,10] instances

		C_{max}		time	
		ratio	**RE(%)**	**ratio**	**RE(%)**
MRHP_FJOB(2window) - MIP	**avg.**	1.165	16.46	0.233	-76.75
	min.	1.048	4.83	0.003	-99.72
	max.	1.364	36.43	1.000	0.00
MRHP_FJOB(3window) - MIP	**avg.**	1.209	20.90	0.226	-77.44
	min.	1.056	5.56	0.003	-99.73
	max.	1.376	37.62	0.957	-4.35
MRHP_BMCH(2window) - MIP	**avg.**	1.147	14.67	0.233	-76.71
	min.	1.028	2.78	0.003	-99.73
	max.	1.317	31.72	1.000	0.00
MRHP_BMCH(3window) - MIP	**avg.**	1.200	20.04	0.231	-76.89
	min.	1.005	0.55	0.003	-99.73
	max.	1.441	44.14	1.000	0.00
MRHP_BJOB(2window) - MIP	**avg.**	1.213	21.25	0.234	-76.56
	min.	1.022	2.16	0.003	-99.73
	max.	1.434	43.41	0.957	-4.35
MRHP_BJOB(3window) - MIP	**avg.**	1.222	22.17	0.216	-78.39
	min.	1.028	2.78	0.003	-99.74
	max.	1.436	43.56	0.913	-8.70
MRHP_FJOB(2window) - MSBH	**avg.**	1.066	6.62	2.013	101.31
	min.	0.957	-4.26	1.846	84.62
	max.	1.185	18.54	2.091	109.09
MRHP_FJOB(3window) - MSBH	**avg.**	1.108	10.82	1.946	94.62
	min.	0.958	-4.22	1.750	75.00
	max.	1.331	33.10	2.364	136.36
MRHP_BMCH(2window) - MSBH	**avg.**	1.050	4.98	2.013	101.26
	min.	0.910	-9.04	1.692	69.23
	max.	1.261	26.06	2.417	141.67
MRHP_BMCH(3window) - MSBH	**avg.**	1.098	9.84	1.997	99.70
	min.	0.947	-5.32	1.769	76.92
	max.	1.307	30.70	2.091	109.09
MRHP_BJOB(2window) - MSBH	**avg.**	1.108	10.80	2.048	104.85
	min.	0.954	-4.61	1.769	76.92
	max.	1.247	24.66	2.917	191.67
MRHP_BJOB(3window) - MSBH	**avg.**	1.120	12.00	1.870	87.04
	min.	0.898	-10.24	1.462	46.15
	max.	1.381	38.10	2.250	125.00
MRHP_FJOB(3window) - MRHP_FJOB(2window)	**avg.**	1.041	4.12	0.966	-3.38
	min.	0.909	-9.09	0.875	-12.50
	max.	1.198	19.76	1.130	13.04
MRHP_BMCH(3window) - MRHP_BMCH(2window)	**avg.**	1.052	5.17	0.995	-0.55
	min.	0.877	-12.27	0.828	-17.24
	max.	1.342	34.23	1.045	4.55
MRHP_BJOB(3window) - MRHP_BJOB(2window)	**avg.**	1.014	1.44	0.916	-8.38
	min.	0.811	-18.92	0.771	-22.86
	max.	1.310	30.99	1.000	0.00

Table 6.21 gives the summary of the results for all the job-machine combinations for the MSBH and the MIP model. From this table, for the MSBH, we see that the average ratio is 1.058 and the average solution gap is 5.78%, and the procedure performs quite well for the 3x10x[11,20] instances (average solution gap is 1.95%). The MSBH needs less time to solve the problems. The average time gap is -80%. As it is seen in Tables B3 and B4 in Appendix B, the average time taken to solve a problem ranges from 11 seconds for 5x3x[4,6] problems with average 25 operations, to 16 seconds for 3x10x[11,30] problems with average 60 operations.

The MSBH produces optimum value for 22 out of 160 problems. Analysis of Tables B1-B8 in Appendix B reveals that 22 (13.8%) of the test problems are solved to optimality with the MSBH, whereas 117 (73.1%) of them are solved to within 10% of optimality, 20 (12.5%) of them are solved to within 20% of optimality and 1 of them is solved to over 20% of optimality. In any case the largest solution gap is only 22.96%. On the other hand, there also exist job-machine sets (3x5x[6,30], 3x5x[6,40] and 5x5x[6,10]) which the MSBH can not find the optimal solution for any instances.

Further analysis of Table 6.21 reveals that, as it is expected, when the number of jobs held constant and the number of machines increases, computational times increase as well. The MSBH methods found the solutions in 9-17 seconds, whereas the branch and bound algorithm solved by LINGO found the optimal solutions of the instances with great computational efforts.

Table 6.21 Summary of the MSBH – MIP comparisons for all job-machine combinations

	C_{max}					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	1.067	6.74	1.008	0.75	1.126	12.59
3x5x[6,40]	1.068	6.81	1.009	0.88	1.173	17.30
3x10x[11,20]	1.020	1.95	1.000	0.00	1.085	8.54
3x10x[11,30]	1.026	2.62	1.000	0.00	1.052	5.24
5x3x[4,6]	1.050	5.02	1.000	0.00	1.160	15.96
5x3x[4,9]	1.074	7.35	1.000	0.00	1.178	17.76
5x5x[6,8]	1.062	6.25	1.000	0.00	1.126	12.59
5x5x[6,10]	1.095	9.47	1.020	2.04	1.230	22.96
avg.	1.058	5.78	1.005	0.46	1.141	14.12
	time					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	0.109	-89.14	0.010	-99.00	0.212	-78.85
3x5x[6,40]	0.047	-95.33	0.001	-99.91	0.119	-88.07
3x10x[11,20]	0.614	-38.585	0.317	-68.29	0.923	-7.692
3x10x[11,30]	0.277	-72.34	0.181	-81.90	0.333	-66.67
5x3x[4,6]	0.158	-84.16	0.007	-99.28	0.550	-45.00
5x3x[4,9]	0.011	-98.87	0.000	-99.96	0.092	-90.76
5x5x[6,8]	0.220	-78.05	0.060	-93.96	0.857	-14.29
5x5x[6,10]	0.114	-88.64	0.001	-99.86	0.478	-52.17
avg.	0.200	-80.04	0.072	-92.77	0.497	-50.31

Table 6.22 and 6.23 gives the summary of the results for all the job-machine combinations for the MSBH_TML and the MSBH_TOTINF with optimal results. Analysis of the tables reveals that the average solution gap of all problems from optimum is 5.84% for the MSBH_TML, and 6.07% for the MSBH_TOTINF. The average of computational time ratios is 0.177 for the MSBH_TML, and 0.273 for the MSBH_TOTINF. As a result the MSBH_TML outperforms the MSBH_TOTINF on both solution quality and time.

Table 6.22 Summary of the MSBH_ TML – the MIP model comparisons for all job-machine combinations

	C_{max}					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	1.069	6.87	1.008	0.75	1.148	14.79
3x5x[6,40]	1.071	7.10	1.007	0.72	1.148	14.81
3x10x[11,20]	1.016	1.58	1.000	0.00	1.060	6.03
3x10x[11,30]	1.019	1.88	1.000	0.00	1.092	9.21
5x3x[4,6]	1.059	5.86	1.000	0.00	1.160	15.96
5x3x[4,9]	1.062	6.18	1.000	0.00	1.178	17.76
5x5x[6,8]	1.076	7.62	1.000	0.00	1.242	24.17
5x5x[6,10]	1.096	9.61	1.020	2.04	1.230	22.96
avg.	1.058	5.84	1.004	0.44	1.157	15.71
	time					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	0.103	-89.69	0.010	-99.00	0.212	-78.85
3x5x[6,40]	0.048	-95.20	0.001	-99.91	0.121	-87.88
3x10x[11,20]	0.595	-40.55	0.293	-70.73	1.111	11.11
3x10x[11,30]	0.219	-78.05	0.143	-85.71	0.280	-72.00
5x3x[4,6]	0.136	-86.45	0.006	-99.41	0.476	-52.38
5x3x[4,9]	0.011	-98.94	0.000	-99.96	0.084	-91.60
5x5x[6,8]	0.198	-80.22	0.058	-94.19	0.786	-21.43
5x5x[6,10]	0.106	-89.39	0.001	-99.88	0.435	-56.52
avg.	0.177	-82.31	0.064	-93.60	0.438	-56.19

Table 6.23 Summary of the MSBH_TOTINF – the MIP model comparisons for all job-machine combinations

	C_{max}					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	1.066	6.55	1.019	1.92	1.125	12.54
3x5x[6,40]	1.068	6.81	1.009	0.88	1.129	12.95
3x10x[11,20]	1.018	1.82	1.000	0.00	1.060	6.03
3x10x[11,30]	1.015	1.48	1.000	0.00	1.046	4.62
5x3x[4,6]	1.058	5.77	1.000	0.00	1.197	19.67
5x3x[4,9]	1.076	7.56	1.000	0.00	1.178	17.76
5x5x[6,8]	1.081	8.12	1.173	17.31	1.000	0.00
5x5x[6,10]	1.105	10.46	1.020	2.04	1.240	24.03
avg.	1.061	6.074	1.028	2.769	1.122	12.201
	time					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	0.127	-87.25	0.014	-98.61	0.250	-75.00
3x5x[6,40]	0.064	-93.64	0.001	-99.87	0.172	-82.83
3x10x[11,20]	0.996	-0.37	0.488	-51.22	1.889	88.89
3x10x[11,30]	0.492	-50.78	0.308	-69.23	0.640	-36.00
5x3x[4,6]	0.160	-83.99	0.007	-99.28	0.550	-45.00
5x3x[4,9]	0.011	-98.86	0.000	-99.96	0.092	-90.76
5x5x[6,8]	0.218	-78.24	0.064	-93.60	0.857	-14.29
5x5x[6,10]	0.114	-88.60	0.001	-99.87	0.522	-47.83
avg.	0.273	-72.716	0.110	-88.956	0.621	-37.851

The comparison of the results for all the job-machine combinations for the MSBH_TML and the MSBH_TOTINF with the MSBH are given in Tables 6.24 and 6.25. Analysis of tables reveals that the average solution gap of all problems from the MSBH is 0.09% for the MSBH_TML, and 0.31% for the MSBH_TOTINF. The average of computational times ratio is 0.91 for the MSBH_TML, and 1.23 for the MSBH_TOTINF. As a result, the MSBH_TML outperforms the MSBH on solution times. Solution times for the MSBH_TOTINF are longer than the MSBH because of the infeasibility profile calculations. Since the MSBH is a sequencing based algorithm, it does not spend too much time in subproblem solving. Thus, in general, its computation time at each iteration is shorter than the MSBH_TOTINF.

Table 6.24 Summary of MSBH_TML – MSBH comparisons for all job-machine combinations

	C_{max}					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	1.001	0.13	0.957	-4.29	1.047	4.69
3x5x[6,40]	1.003	0.29	0.953	-4.71	1.053	5.32
3x10x[11,20]	0.997	-0.35	0.974	-2.55	1.025	2.53
3x10x[11,30]	0.993	-0.70	0.950	-4.98	1.069	6.87
5x3x[4,6]	1.008	0.84	0.986	-1.41	1.076	7.63
5x3x[4,9]	0.989	-1.05	0.904	-9.60	1.022	2.24
5x5x[6,8]	1.013	1.33	0.959	-4.10	1.155	15.50
5x5x[6,10]	1.002	0.23	0.910	-9.04	1.105	10.48
avg.	1.001	0.090	0.949	-5.086	1.069	6.906
	time					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	0.939	-6.11	0.800	-20.00	1.000	0.00
3x5x[6,40]	1.026	2.63	0.833	-16.67	1.154	15.38
3x10x[11,20]	0.907	-9.33	0.833	-16.67	1.000	0.00
3x10x[11,30]	0.798	-20.19	0.688	-31.25	1.000	0.00
5x3x[4,6]	0.849	-15.13	0.692	-30.77	0.909	-9.09
5x3x[4,9]	0.934	-6.60	0.833	-16.67	1.000	0.00
5x5x[6,8]	0.897	-10.26	0.769	-23.08	1.000	0.00
5x5x[6,10]	0.921	-7.94	0.833	-16.67	1.000	0.00
avg.	0.909	-9.117	0.785	-21.470	1.008	0.787

Table 6.25 Summary of MSBH_TOTINF – MSBH comparisons for all job-machine combinations

	C_{max}					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	0.999	-0.14	0.945	-5.50	1.059	5.85
3x5x[6,40]	1.000	0.02	0.957	-4.29	1.041	4.05
3x10x[11,20]	0.999	-0.10	0.935	-6.47	1.047	4.71
3x10x[11,30]	0.989	-1.09	0.954	-4.58	1.020	1.98
5x3x[4,6]	1.008	0.78	0.915	-8.45	1.106	10.61
5x3x[4,9]	1.002	0.21	0.904	-9.60	1.073	7.28
5x5x[6,8]	1.018	1.79	0.959	-4.10	1.085	8.53
5x5x[6,10]	1.010	1.02	0.901	-9.87	1.120	11.97
avg.	1.003	0.312	0.934	-6.608	1.069	6.872
	time					
	average		**minimum**		**maximum**	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
3x5x[6,30]	1.194	19.42	1.083	8.33	1.385	38.46
3x5x[6,40]	1.383	38.31	1.231	23.08	1.750	75.00
3x10x[11,20]	1.518	51.76	1.385	38.46	1.833	83.33
3x10x[11,30]	1.789	78.92	1.333	33.33	2.579	157.89
5x3x[4,6]	1.009	0.85	0.846	-15.38	1.100	10.00
5x3x[4,9]	1.006	0.63	0.846	-15.38	1.091	9.09
5x5x[6,8]	0.986	-1.42	0.846	-15.38	1.091	9.09
5x5x[6,10]	0.985	-1.53	0.846	-15.38	1.091	9.09
avg.	1.234	23.369	1.052	5.208	1.490	48.995

Table 6.26 presents the summary of all the results. Analysis of Table 6.26 reveals that the solution gap between the MSBH and its two versions with the MIP method is 5%-6% in average. But, they produce these results in 72%-82% less computational times. The solution gap between the MSBH_TML, the MSBH_TOTINF with the MSBH is less than 1%. The MSBH_TML perform slightly better than the MSBH_TOTINF. The MSBH_TML outperforms both the MSBH_TOTINF and the MSBH on solution time, whereas the MSBH outperforms the MSBH_TOTINF. In any case the largest solution gap from optimum is 16% which is obtained by MRHP_BMCH(window3). Further analysis of Table 6.26 reveals that the MSBH outperforms all the MRHP methods. The MHRP methods with two time windows outperform the methods with three time windows on solution quality, whereas they are outperformed on solution time.

Table 6.26 Summary of all the results for small size test problems

	C_{max}		time	
	ratio	**RE(%)**	**ratio**	**RE(%)**
MSBH - MIP	1.058	5.78	0.199	-80.08
MSBH_TML - MIP	1.058	5.84	0.177	-82.31
MSBH_TOTINF - MIP	1.061	6.07	0.273	-72.72
MSBH_TML - MSBH	1.001	0.09	0.913	-8.72
MSBH_TOTINF - MSBH	1.003	0.31	1.241	24.11
MSBH_TOTINF – MSBH_TML	1.003	0.26	1.374	37.39
MRHP_FJOB(2window) - MIP	1.111	11.07	0.411	-58.93
MRHP_FJOB(3window) - MIP	1.149	14.92	0.398	-60.25
MRHP_BMCH(2window) - MIP	1.123	12.32	0.411	-58.88
MRHP_BMCH(3window) - MIP	1.162	16.19	0.391	-60.89
MRHP_BJOB(2window) - MIP	1.131	13.13	0.417	-58.33
MRHP_BJOB(3window) - MIP	1.157	15.71	0.393	-60.72
MRHP_FJOB(2window) - MSBH	1.051	5.05	2.064	106.36
MRHP_FJOB(3window) - MSBH	1.087	8.71	1.995	99.46
MRHP_BMCH(2window) - MSBH	1.062	6.23	2.059	105.86
MRHP_BMCH(3window) - MSBH	1.099	9.91	2.001	100.12
MRHP_BJOB(2window) - MSBH	1.069	6.94	1.652	65.23
MRHP_BJOB(3window) - MSBH	1.094	9.44	1.982	98.20
MRHP_FJOB(3window) - MRHP_FJOB(2window)	1.036	3.63	0.968	-3.21
MRHP_BMCH(3window) - MRHP_BMCH(2window)	1.038	3.84	0.969	-3.08
MRHP_BJOB(3window) - MRHP_BJOB(2window)	1.026	2.64	0.956	-4.37

It was shown that the MSBH generally outperforms the MSBH_TML and the MSBH_TOTINF on solution quality. In terms of solution quality, the MSBH_TML appears to perform slightly better than the MSBH_TOTINF, but the difference is not significant.

For the small size problem instances, the MSBH could find quite better solutions than the MRHP does. All the MSBH versions significantly outperform the MRHP versions. As it is expected, decomposition of the problem by the MHRP and then solving each subproblem separately by the MSBH_TML prevents the evaluation all the disjunctive graph of the instance. So, the MSBH_TML can not perform a good schedule over all, whereas it produces good results for each subproblem.

As described above, two cases were considered for the MRHP; the MRHP with two time windows and the MRHP with three time windows. The impact of the number of time windows considered in the decomposition on the makespan as well as the time needed to obtain the solution was studied. As it is expected, when the

number of time windows increases, the quality and solution time decreases. The first expectation is that the quality of the solution decreases when the number of time windows involved in the decomposition increases. This was confirmed in most tests. Even if the first expectation is that C_{max} increases when the number of time windows is greater, in some cases, the solution obtained with three time windows is better than the one obtained with two time windows. It is clear and expectable that the CPU time decreases when the number of time windows increases.

Similarly, experiments show that the MRHP might retrieve, in few cases, a better solution than the one found by the MSBH. These exceptions are due to the fact that the SBH is a heuristic for an NP-hard problem; it behaves efficiently in practice, but for some problems, it might give a better solution with a decomposition heuristic than with a classical SBH.

Finally, we compared the results to evaluate the differences using the paired t-test at the 1% level of significance. Table 6.27 and 6.28 present summary of all the results compared with the MIP model and the MSBH successively. As it expected, the difference of the results of the proposed methods and the MIP model are statistically significant. The proposed methods have solved the problems in less computational times but the solutions qualities are not as well as the MIP model. In Table 6.28, the results whose difference from the MSBH is not statistically significant are highlighted in bold type. The difference of the results of the MSBH_TML and the MSBH_TOTINF with the MSBH are not statistically significant on solution quality, whereas it is significant on solution time. Furthermore, the difference of the results of the MSBH_TML and the MSBH_TOTINF is not statistically significant on solution quality, whereas it is significant on solution time. The comparison of the MRHP methods for different number of time windows are presented in Table 6.29. Analysis of Table 6.29 reveals that the difference of the results of the MRHP_BMCH(3window) and the MRHP_BMCH(2window) are not statistically significant on both solution quality and time.

Table 6.27 Solution time and quality ratios for all instances with respect to the MIP model

	C_{max}	time
MSBH	1.058	0.199
MSBH_TML	1.058	0.177
MSBH_TOTINF	1.061	0.273
MRHP_FJOB(2window)	1.111	0.411
MRHP_FJOB(3window)	1.149	0.398
MRHP_BMCH(2window)	1.123	0.411
MRHP_BMCH(3window)	1.162	0.391
MRHP_BJOB(2window)	1.131	0.417
MRHP_BJOB(3window)	1.157	0.393

Table 6.28 Solution time and quality ratios for all instances with respect to the MSBH

	C_{max}	time
MSBH_TML	**1.001**	0.913
MSBH_TOTINF	**1.003**	1.241
MSBH_TOTINF - MSBH_TML	**1.003**	1.374
MRHP_FJOB(2window)	1.051	2.064
MRHP_FJOB(3window)	1.087	1.995
MRHP_BMCH(2window)	1.062	2.059
MRHP_BMCH(3window)	1.099	2.001
MRHP_BJOB(2window)	1.069	1.652
MRHP_BJOB(3window)	1.094	1.982

Table 6.29 Solution time and quality ratios for all instances for different number of time windows.

	C_{max}	time
MRHP_FJOB(3window) - MRHP_FJOB(2window)	1.036	0.968
MRHP_BMCH(3window) - MRHP_BMCH(2window)	**1.038**	**0.969**
MRHP_BJOB(3window) - MRHP_BJOB(2window)	**1.026**	0.956

Another potential contributor is the effect of the resequencing step. It is quite possible that the resequencing procedure is capable of offsetting the poorer solutions that might result from the MSBH. In our second set of experiments we explore the effects of using different number of cycles in the resequencing step. The CPU time increases as the number of cycles increases, and the performance becomes better, which is to be expected. Incorporating the resequencing step with six cycles increased the heuristic's solution time by an average of 266%.

Tables 6.30 and 6.31 displays successively the average computation times and average C_{max} values for each combination of the number of jobs and machines, and resequencing methodology.

Table 6.30 The effect of resequencing on computation time for the MSBH

	No resequencing	1 cycle	2 cycle	3 cycle	4 cycle	5 cycle	Full resequencing (6 cycle)
3x5x[6,30]	3.4	11.7	12.2	12.2	12.2	12.2	12.2
3x5x[6,40]	3.7	11.8	12.2	12.3	12.3	12.3	12.3
3x10x[11,20]	3.1	12.5	12.5	12.5	12.5	12.5	12.5
3x10x[11,30]	3.2	14.3	15.0	15.0	15.0	15.0	15.0
5x3x[4,6]	2.9	10.3	10.7	10.9	10.9	11.0	11.0
5x3x[4,9]	3.0	10.6	11.0	11.1	11.1	11.1	11.1
5x5x[6,8]	3.5	10.4	11.3	11.4	11.4	11.4	11.4
5x5x[6,10]	3.7	10.5	11.3	11.4	11.6	11.6	11.6

Table 6.31 The effect of resequencing on solution quality for the MSBH

	No resequencing	1 cycle	2 cycle	3 cycle	4 cycle	5 cycle	Full resequencing (6 cycle)
3x5x[6,30]	309.00	296.25	296.25	296.25	296.25	296.25	296.25
3x5x[6,40]	381.20	361.25	360.35	360.35	360.35	360.35	360.35
3x10x[11,20]	206.70	201.20	201.20	201.20	201.20	201.20	201.20
3x10x[11,30]	297.50	287.90	287.90	287.90	287.90	287.90	287.90
5x3x[4,6]	123.30	117.00	116.95	116.95	116.95	116.95	116.75
5x3x[4,9]	166.45	155.35	154.65	154.65	154.65	154.65	154.65
5x5x[6,8]	127.35	120.25	120.20	120.20	120.20	120.20	120.20
5x5x[6,10]	159.85	150.35	150.35	150.35	150.35	150.35	150.35

To clarify this issue we examined the relationship between the number of resequencing cycles and the solution quality. It appears that while for some instances good solutions can be obtained with no resequencing, in many others the solution quality depends on the amount of resequencing that takes place. However, for a substantial fraction of instances solution quality improves with resequencing.

The results for MSBH(NR) and MSBH(FR) highlight the importance of the resequencing task. The MSBH with different resequencing cycles consistently outperforms MSBH(NR) in solution quality. Thus, resequencing is very important to achieving high quality solutions. However, MSBH(NR) requires only 27% of the time taken by MSBH(FR). Thus, it is possible to control the tradeoff between solution quality and computation time by controlling the amount of resequencing.

After the experimental test, a sensitivity analysis is conducted on the average number of operations. Table 6.32 shows the average number of operations per job, and the average solution times obtained by the MSBH for each problem set. Based

on the results, it can be seen that the relative performance of the MSBH on solution time is sensitive to changes in the number of operations. If we hold the number of jobs and machines constant and increase the number of operations, computation times increase. For example 3x10x[11,20] problems find the solution in 12.5 seconds in average, whereas 3x10x[11,30], which has the same number of jobs and machines with 3x10x[11,20] instances but has more operations, solve the problems in 15.0 seconds. If we hold the number of jobs and maximum number of operations per job constant and increase the number of machines, computation times increase, too. For example, 3x5x[6,30] and 3x10x[11,30] problem sets have the same number of jobs and maximum number of operations, whereas the first one has 5 machines and the second one has 10 machines. Analysis of Table 6.32 reveals that the average computation time for 3x5x[6,30] set is 12.2 seconds, whereas it is 15.0 seconds for 3x10x[11,30] problem set. As the number of machines increases, the computational time increases. Furthermore, increase in the number of machines increases the solution time more than the increase in the number of jobs.

Figures 6.10, 6.11, 6.12 and 6.13 present the solution times of all the instances for four different job-machine combinations, in order to show the impact of number of operations on solution times. We can see the trend of the increase on solution times when the number of operations of the instances are increased. Especially, the trend can be seen in 3 jobs and 5 or 10 machines instances in Figures 6.10 and 6.11.

Table 6.32 The number of operations and solution times

	Average number of operations per job	**Average computation time for MSBH**
3x5x[6,30]	51.8	12.2
3x5x[6,40]	64.1	12.3
3x10x[11,20]	43.8	12.5
3x10x[11,30]	59.2	15.0
5x3x[4,6]	24.7	11.0
5x3x[4,9]	33.3	11.1
5x5x[6,8]	34.9	11.4
5x5x[6,10]	39.2	11.6

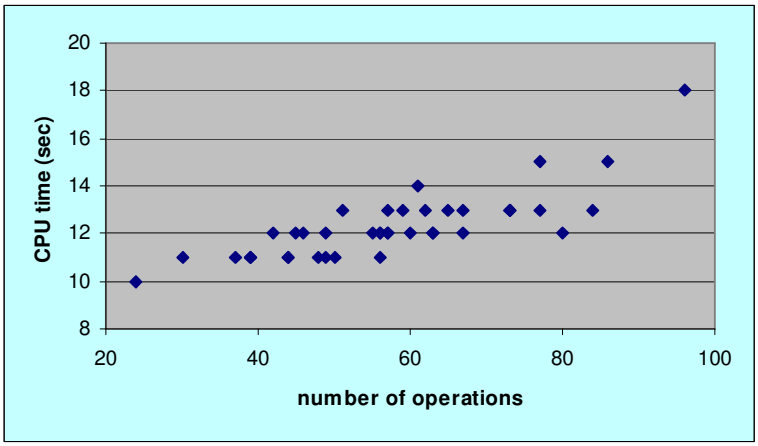


Figure 6.10 The solution times for the 3 jobs 5 machine instances

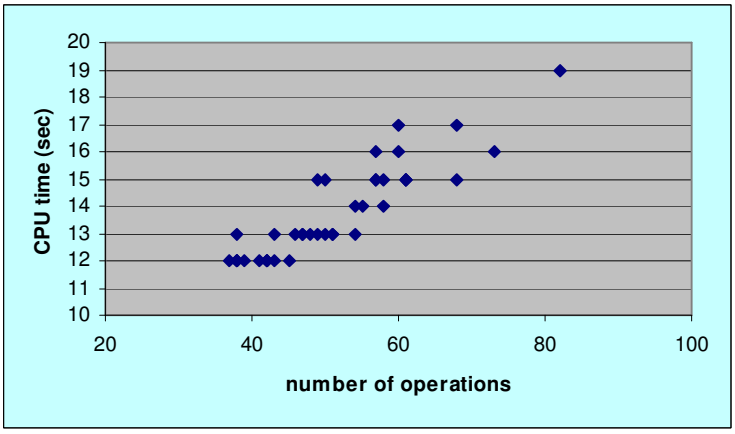


Figure 6.11 The solution times for the 3 jobs 10 machine instances

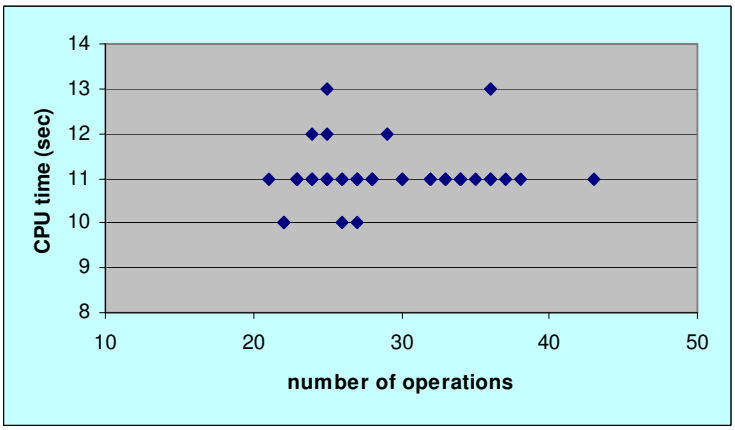


Figure 6.12 The solution times for the 5 jobs 3 machines instances

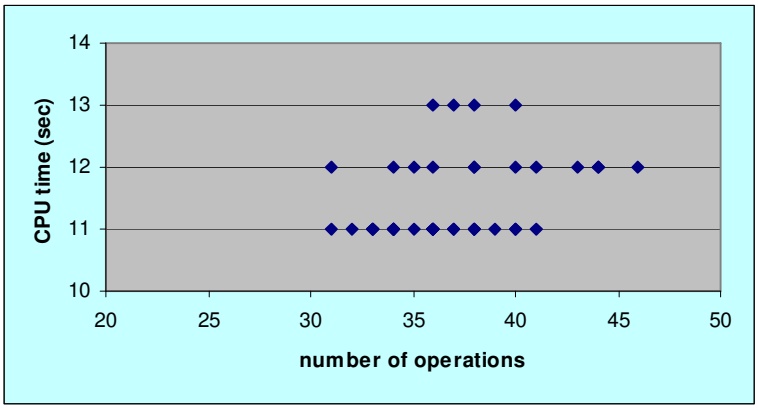


Figure 6.13 The solution times for the 5 jobs 5 machines instances

6.4 Comparison Of The Proposed Methods For The Reentrant Job Shop Scheduling Problem For Large Test Problems

In the previous section, we can compare the proposed methods with optimum using small size instances. In this section, we enlarge the size of problems, and investigate the performance of the proposed methods on larger problems from two aspects: makespan and CPU time. Of course, we can not solve these instances as to find optimum. For example, for a 3x5x[6,40] problem, we can not find the optimum solution over a twenty five hours by LINGO.

We use an extensive set of large randomly generated test problems which allows us to link algorithms performance to problem characteristics. The problems are tested on both the MSBH and its two versions with the MSBH_TML and the MSBH_TOTINF. Furthermore, the MRHP with different number of time windows and different rules for assignment of operations to time windows are applied to the test problems.

We consider six levels for the number of jobs (n = 10, 15, 20, 30, 50, 100) and four levels for the number of machines (m = 5, 10, 15, 20). All jobs are available simultaneously at time zero and the processing times are randomly generated from a discrete uniform distribution over [1,20]. The sizes of the problems are as follows:

- 10 x 5 x [6,10]
- 10 x 10 x [11,15]
- 15 x 10 x [11,15]
- 15 x 15 x [16,20]
- 20 x 15 x [16,20]
- 20 x 20 x [21,25]
- 30 x 15 x [16,20]
- 30 x 20 x [21,25]
- 50 x 15 x [16,20]
- 50 x 20 x [21,25]
- 100 x 15 x [16,20]
- 100 x 20 x [21,25]

For each of the twelve job-machine combinations, we generate 10 instances. Thus our experimental design consist of 120 randomly generated test problems. The maximum number of operations of a job in these problems is 2326, and the minimum number of operations is 70.

The number of resequencing cycles is fixed to one to shorten the computation times. We see in Section 7.2 that more than one cycle does not significantly reduce the makespan. The MRHP is applied with two and three time windows. The three rules (MRHP_FJOB, MRHP_BMCH, MRHP_BJOB) for assignment of operations to time windows are used for the MRHP.

6.4.1 Details Of The Implementation And The Computational Results

The objectives of this experiment are twofold: to evaluate the performance of the MRHP methods relative to the MSBH, the MSBH_TML and the MSBH_TOTINF for large makespan problems. Furthermore, 120 large problems are tested to determine what size problem can be addressed using the MSBH in reasonable solution times.

All the results of this experiment are shown in Appendix B. A summary of these results are given in Tables 6.33, 6.34, 6.35 and 6.36 for all job-machine combinations. Table 6.33 shows the solution quality ratios relative to C_{max} values obtained by the MSBH, the MSBH_TML and the MSBH_TOTINF. To compare the computation times of the methods, we calculate average computation times over all instances in a problem class. Table 6.34 shows the results of solution qualities and computation times for the MHRP with two time windows. Table 6.35 shows the results for the MHRP with three time windows. Table 6.36 shows the comparisons of the MRHP with two and three time windows.

In the remainder of this section, the results are evaluated via averages of all job-machine combinations. To illustrate the results of a job machine combination, the results of 10x5x[6,10] problems are summarized below. For example, for the 10x5x[6,10] instances, the MSBH produces the lowest C_{max} schedules in 6 of 10 case investigated as it is seen in Table B9 in Appendix B. Figure 6.14 shows the results of all the MSBH methods for the 10x5x[6,10] problems. Figure 6.15 shows the results of all the MRHP methods for the 10x5x[6,10] problems. The MSBH procedures significantly outperform the MRHP over the 10x5x[6,10] problem set. As it is seen in Appendix B, one of the MRHP methods found the lower C_{max} value than the MSBH for only 3 of the 10 instances. Furthermore, for 10x5x[6,10] instances, because the number of jobs not too much, the MSBH methods require less computational times than the MRHP.

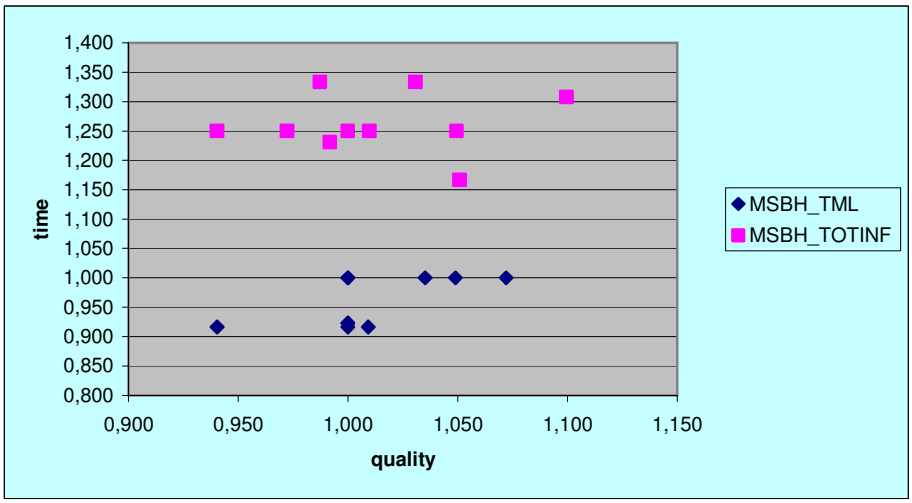


Figure 6.14 The results for the MSBH methods for the 10x5x[6,10] instances.

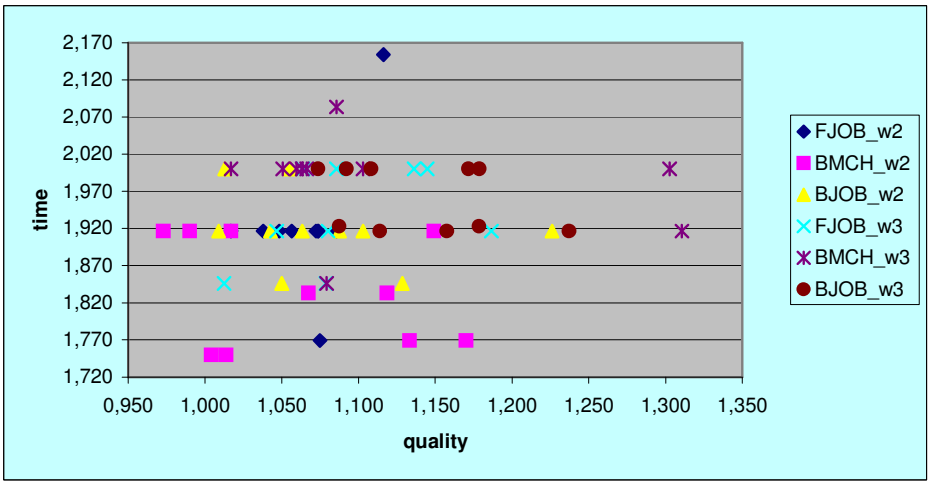


Figure 6.15 The results for the MRHP for the 10x5x[6,10] instances.

Table 6.33 Comparison of the MSBH, the MSBH_TML and the MSBH_TOTINF

		MSBH		MSBH_TML		MSBH_TOTINF		MSBH_TML - MSBH comparison				MSBH_TOTINF - MSBH comparison				MSBH_TOTINF - MSBH_TML comparison			
		C_{max}	time	C_{max}	time	C_{max}	time	C_{max}		time		C_{max}		time		C_{max}		time	
								ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
10x5x[6.10]	**avg**	223.6	12.2	225.8	11.7	226.4	15.4	1.011	1.06	0.960	-4.04	1.013	1.32	1.262	26.22	1.003	0.30	1.317	31.67
	min	177.0	12.0	177.0	11.0	186.0	14.0	0.940	-5.96	0.917	-8.33	0.940	-5.96	1.167	16.67	0.963	-3.74	1.250	25.00
	max	241.0	13.0	241.0	12.0	265.0	17.0	1.072	7.21	1.000	0.00	1.100	9.96	1.333	33.33	1.100	9.96	1.417	41.67
10x10x[11.15]	**avg**	261.6	24.1	266.7	22.4	255.9	44.4	1.022	2.24	0.930	-6.99	0.981	-1.90	1.844	84.44	0.960	-3.96	1.985	98.45
	min	227.0	23.0	236.0	21.0	222.0	40.0	0.955	-4.45	0.840	-16.00	0.899	-10.12	1.600	60.00	0.893	-10.73	1.800	80.00
	max	296.0	25.0	309.0	25.0	281.0	49.0	1.150	15.00	1.042	4.17	1.084	8.37	2.130	113.04	1.004	0.39	2.130	113.04
15x10x[11.15]	**avg**	346.1	36.8	343.0	32.6	338.4	75.5	0.993	-0.74	0.887	-11.30	0.980	-2.00	2.050	105.00	0.988	-1.21	2.313	131.32
	min	304.0	33.0	321.0	31.0	308.0	68.0	0.935	-6.52	0.842	-15.79	0.915	-8.47	1.971	97.14	0.938	-6.19	2.194	119.35
	max	413.0	41.0	404.0	35.0	379.0	85.0	1.056	5.59	0.939	-6.06	1.027	2.74	2.158	115.79	1.036	3.65	2.471	147.06
15x15x[16.20]	**avg**	370.5	101.1	365.2	88.1	368.8	225.6	0.988	-1.24	0.873	-12.74	0.997	-0.29	2.237	123.71	1.011	1.06	2.563	156.30
	min	350.0	96.0	337.0	83.0	333.0	203.0	0.903	-9.68	0.817	-18.27	0.900	-10.00	2.026	102.63	0.895	-10.48	2.341	134.07
	max	403.0	114.0	395.0	94.0	400.0	242.0	1.129	12.86	0.927	-7.29	1.120	12.00	2.469	146.94	1.089	8.90	2.814	181.40
20x15x[16.20]	**avg**	439.5	171.4	444.9	136.7	433.9	355.0	1.015	1.50	0.798	-20.23	0.989	-1.13	2.071	107.11	0.977	-2.34	2.602	160.25
	min	405.0	160.0	415.0	124.0	401.0	301.0	0.922	-7.78	0.716	-28.41	0.924	-7.60	1.858	85.80	0.914	-8.58	2.298	129.77
	max	487.0	183.0	481.0	152.0	463.0	384.0	1.151	15.06	0.889	-11.11	1.052	5.19	2.244	124.38	1.067	6.68	2.889	188.89
20x20x[21.25]	**avg**	486.4	364.0	480.3	296.8	470.8	802.2	0.988	-1.20	0.816	-18.42	0.968	-3.17	2.205	120.51	0.982	-1.80	2.703	170.34
	min	470.0	330.0	440.0	276.0	442.0	741.0	0.921	-7.91	0.777	-22.28	0.915	-8.49	2.070	106.99	0.885	-11.47	2.607	160.70
	max	509.0	386.0	532.0	313.0	503.0	844.0	1.104	10.37	0.844	-15.63	1.019	1.91	2.307	130.66	1.079	7.94	2.804	180.40

Table 6.33 (Cont'd)

		MSBH		MSBH_TML		MSBH_TOTINF		MSBH_TML - MSBH comparison				MSBH_TOTINF - MSBH comparison				MSBH_TOTINF - MSBH_TML comparison			
		C_{max}	time	C_{max}	time	C_{max}	time	C_{max}		time		C_{max}		time		C_{max}		time	
								ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
30x15x[16.20]	**avg**	572.1	325.9	558.6	255.7	559.5	700.5	0.977	-2.29	0.785	-21.46	0.980	-2.05	2.149	114.93	1.003	0.32	2.739	173.90
	min	528.0	313.0	510.0	245.0	519.0	652.0	0.891	-10.91	0.732	-26.80	0.921	-7.92	2.083	108.31	0.939	-6.11	2.567	156.69
	max	638.0	362.0	638.0	267.0	599.0	761.0	1.034	3.35	0.816	-18.41	1.049	4.92	2.233	123.30	1.061	6.07	2.872	187.17
30x20x[21.25]	**avg**	619.6	769.0	616.0	613.0	623.0	1599.3	0.996	-0.37	0.797	-20.29	1.007	0.69	2.080	107.96	1.012	1.21	2.610	160.98
	min	571.0	747.0	571.0	585.0	581.0	1541.0	0.915	-8.49	0.781	-21.94	0.931	-6.94	2.035	103.47	0.954	-4.60	2.522	152.22
	max	651.0	808.0	639.0	653.0	667.0	1684.0	1.100	9.98	0.847	-15.30	1.090	8.99	2.136	113.62	1.102	10.16	2.721	172.05
50x15x[16.20]	**avg**	827.3	840.6	832.0	650.9	842.4	1760.5	1.006	0.59	0.775	-22.54	1.016	1.63	2.091	109.09	1.011	1.09	2.700	170.04
	min	770.0	774.0	764.0	615.0	776.0	1565.0	0.958	-4.15	0.747	-25.30	0.954	-4.64	1.906	90.62	0.947	-5.33	2.426	142.64
	max	960.0	887.0	967.0	692.0	1085.0	2425.0	1.071	7.10	0.798	-20.16	1.130	13.02	2.734	173.39	1.122	12.20	3.535	253.50
50x20x[21.25]	**avg**	865.6	1971.1	857.4	1515.1	842.3	3840.0	0.993	-0.72	0.769	-23.15	0.976	-2.40	1.948	94.80	0.983	-1.68	2.536	153.58
	min	750.0	1861.0	769.0	1412.0	775.0	3577.0	0.922	-7.79	0.750	-24.96	0.886	-11.36	1.883	88.27	0.937	-6.29	2.398	139.79
	max	942.0	2025.0	910.0	1590.0	909.0	3997.0	1.131	13.07	0.795	-20.50	1.120	12.00	2.007	100.67	1.040	4.00	2.653	165.28
100x15x[16.20]	**avg**	1507.1	3359.9	1492.8	2488.5	1491.6	6345.3	0.990	-0.95	0.741	-25.90	0.990	-1.03	1.890	88.96	0.999	-0.07	2.550	154.98
	min	1390.0	3161.0	1374.0	2345.0	1373.0	5827.0	0.967	-3.26	0.704	-29.61	0.969	-3.06	1.733	73.31	0.990	-1.00	2.462	146.23
	max	1580.0	3620.0	1574.0	2587.0	1574.0	6774.0	0.999	-0.07	0.764	-23.61	1.005	0.54	2.020	101.97	1.006	0.61	2.722	172.16
100x20x[21.25]	**avg**	1504.0	8188.1	1461.5	5893.3	1460.7	14234.3	0.972	-2.78	0.720	-28.03	0.972	-2.84	1.739	73.86	1.000	-0.05	2.416	141.58
	min	1435.0	7806.0	1371.0	5658.0	1375.0	13453.0	0.927	-7.25	0.697	-30.27	0.926	-7.44	1.658	65.80	0.993	-0.67	2.332	133.18
	max	1586.0	8509.0	1538.0	6137.0	1536.0	14892.0	1.007	0.74	0.740	-26.00	1.001	0.07	1.803	80.35	1.007	0.69	2.484	148.42

Table 6.34 Comparison of the MSBH and the MRHP methods with two time windows

		MSBH		MRHP_BMCH 2 time windows		MRHP_FJOB 2 time windows		MRHP_BJOB 2 time windows		MRHP_BMCH- MSBH comparison				MRHP_FJOB - MSBH comparison				MRHP_BJOB - MSBH comparison			
										C_{max}		time		C_{max}		time		C_{max}		time	
		C_{max}	time	C_{max}	time	C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
10x5x[6.10]	avg	223.6	12.2	238.0	22.4	238.3	23.5	240.2	23.4	1.064	6.37	1.837	83.72	1.066	6.56	1.926	92.56	1.078	7.78	1.919	91.92
	min	177.0	12.0	198.0	21.0	187.0	23.0	217.0	23.0	0.973	-2.70	1.750	75.00	1.017	1.70	1.769	76.92	1.009	0.90	1.846	84.62
	max	241.0	13.0	282.0	23.0	269.0	28.0	272.0	24.0	1.170	17.01	1.917	91.67	1.116	11.62	2.154	115.38	1.226	22.60	2.000	100.0
10x10x[11.15]	avg	261.6	24.1	270.6	37.1	278.0	36.0	280.7	37.0	1.037	3.67	1.540	53.97	1.063	6.33	1.495	49.50	1.075	7.53	1.537	53.69
	min	227.0	23.0	234.0	35.0	228.0	34.0	241.0	35.0	0.921	-7.90	1.440	44.00	0.950	-5.00	1.400	40.00	0.976	-2.43	1.400	40.00
	max	296.0	25.0	295.0	44.0	313.0	39.0	312.0	39.0	1.115	11.45	1.760	76.00	1.145	14.54	1.625	62.50	1.185	18.50	1.625	62.50
15x10x[11.15]	avg	346.1	36.8	354.3	55.1	364.3	50.4	370.0	53.5	1.025	2.51	1.505	-83.90	1.053	5.33	1.372	37.19	1.069	6.92	1.452	45.19
	min	304.0	33.0	334.0	49.0	304.0	48.0	328.0	48.0	0.995	-0.54	1.282	-87.17	0.918	-8.16	1.268	26.83	0.991	-0.91	1.371	37.14
	max	413.0	41.0	422.0	65.0	434.0	54.0	446.0	66.0	1.099	9.87	1.970	-78.62	1.148	14.80	1.455	45.45	1.155	15.48	1.610	60.98
15x15x[16.20]	avg	370.5	101.1	406.7	123.1	389.7	113.0	395.4	125.5	1.099	9.92	1.220	21.97	1.054	5.37	1.119	11.91	1.069	6.86	1.244	24.44
	min	350.0	96.0	351.0	119.0	368.0	108.0	368.0	118.0	0.921	-7.87	1.096	9.65	0.982	-1.84	1.067	6.67	0.969	-3.15	1.105	10.53
	max	403.0	114.0	467.0	129.0	420.0	123.0	447.0	136.0	1.272	27.25	1.283	28.28	1.163	16.29	1.198	19.79	1.252	25.21	1.360	36.00
20x15x[16.20]	avg	439.5	171.4	464.8	175.5	464.0	171.1	472.3	183.0	1.059	5.87	1.025	2.45	1.058	5.83	0.998	-0.15	1.075	7.49	1.067	6.75
	min	405.0	160.0	410.0	164.0	427.0	160.0	401.0	164.0	0.949	-5.13	0.960	-3.98	0.922	-7.80	0.909	-9.09	0.990	-0.99	1.012	1.23
	max	487.0	183.0	505.0	192.0	507.0	187.0	531.0	203.0	1.128	12.76	1.091	9.09	1.151	15.07	1.028	2.84	1.232	23.20	1.153	15.34
20x20x[21.25]	avg	486.4	364.0	498.1	397.9	484.1	373.3	514.5	399.0	1.025	2.46	1.093	9.35	0.996	-0.40	1.027	2.73	1.058	5.83	1.098	9.77
	min	470.0	330.0	475.0	354.0	452.0	353.0	477.0	376.0	0.939	-6.13	0.982	-1.81	0.898	-10.22	0.948	-5.18	0.996	-0.42	0.977	-2.33
	max	509.0	386.0	560.0	432.0	513.0	396.0	554.0	434.0	1.109	10.89	1.158	15.82	1.085	8.46	1.142	14.24	1.122	12.24	1.215	21.49

Table 6.34 (Cont'd)

		MSBH		MRHP_BMCH 2 time windows		MRHP_FJOB 2 time windows		MRHP_BJOB 2 time windows		MRHP_BMCH- MSBH comparison				MRHP_FJOB - MSBH comparison				MRHP_BJOB - MSBH comparison			
										C_{max}		time		C_{max}		time		C_{max}		time	
		C_{max}	time	C_{max}	time	C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
30x15x[16.20]	avg	572.1	325.9	587.9	337.6	575.9	323.6	595.8	344.1	1.030	3.00	1.034	3.36	1.009	0.88	0.992	-0.76	1.04	4.278	1.06	5.7
	min	528.0	313.0	503.0	300.0	536.0	305.0	559.0	326.0	0.901	-9.90	0.952	-4.76	0.909	-9.06	0.936	-6.44	0.97	-3.356	0.98	-2.2
	max	638.0	362.0	637.0	417.0	619.0	382.0	657.0	376.0	1.160	16.03	1.152	15.19	1.138	13.83	1.055	5.52	1.12	11.732	1.17	17.1
30x20x[21.25]	avg	619.6	769.0	634.4	808.1	633.8	728.7	647.2	788.4	1.025	2.54	1.050	5.02	1.024	2.38	0.948	-5.23	1.046	4.58	1.025	2.53
	min	571.0	747.0	601.0	739.0	570.0	711.0	613.0	747.0	0.929	-7.10	0.969	-3.15	0.936	-6.41	0.932	-6.84	0.982	-1.76	0.988	-1.16
	max	651.0	808.0	697.0	970.0	702.0	759.0	674.0	834.0	1.119	11.88	1.223	22.32	1.127	12.68	0.962	-3.77	1.101	10.13	1.088	8.76
50x15x[16.20]	avg	827.3	840.6	855.0	773.7	850.7	811.0	863.2	840.3	1.034	3.44	0.921	-7.94	1.029	2.91	0.963	-3.68	1.045	4.49	1.000	0.01
	min	770.0	774.0	795.0	699.0	771.0	719.0	794.0	771.0	0.989	-1.05	0.894	-10.65	0.982	-1.78	0.922	-7.78	0.988	-1.19	0.955	-4.51
	max	960.0	887.0	954.0	831.0	957.0	1016.0	951.0	884.0	1.104	10.39	0.974	-2.64	1.127	12.67	1.149	14.93	1.139	13.90	1.035	3.47
50x20x[21.25]	avg	865.6	1971.1	859.3	1795.2	868.0	1769.3	880.6	1917.5	0.997	-0.35	0.911	-8.94	1.006	0.59	0.898	-10.23	1.021	2.05	0.973	-2.70
	min	750.0	1861.0	812.0	1647.0	807.0	1662.0	833.0	1856.0	0.887	-11.25	0.852	-14.77	0.921	-7.86	0.859	-14.07	0.921	-7.86	0.923	-7.71
	max	942.0	2025.0	915.0	1949.0	963.0	1841.0	959.0	2082.0	1.151	15.07	0.973	-2.74	1.161	16.13	0.928	-7.21	1.120	12.00	1.028	2.81
100x15x[16.20]	avg	1507.1	3359.9	1547.3	2829.0	1579.8	2830.3	1585.5	3087.2	1.027	2.69	0.842	-15.79	1.048	4.84	0.843	-15.68	1.052	5.24	0.920	-8.03
	min	1390.0	3161.0	1398.0	2573.0	1413.0	2743.0	1434.0	2898.0	0.989	-1.08	0.793	-20.66	0.985	-1.53	0.799	-20.11	1.008	0.76	0.858	-14.2
	max	1580.0	3620.0	1603.0	3001.0	1678.0	2917.0	1650.0	3190.0	1.085	8.49	0.891	-10.85	1.094	9.35	0.869	-13.12	1.119	11.94	0.956	-4.40
100x20x[21.25]	avg	1504.0	8188.1	1524.9	6588.8	1548.5	6658.0	1545.2	7126.5	1.014	1.44	0.804	-19.55	1.030	3.04	0.813	-18.66	1.028	2.82	0.870	-12.9
	min	1435.0	7806.0	1429.0	6247.0	1487.0	6441.0	1475.0	6676.0	0.958	-4.17	0.779	-22.06	0.945	-5.53	0.794	-20.62	0.954	-4.60	0.823	-17.7
	max	1586.0	8509.0	1602.0	6989.0	1652.0	6854.0	1693.0	7775.0	1.059	5.88	0.825	-17.50	1.090	8.99	0.848	-15.18	1.087	8.73	0.924	-7.64

Table 6.35 Comparison of the MSBH and the MRHP methods with three time windows

		MSBH		MRHP_BMCH 3 time windows		MRHP_FJOB 3 time windows		MRHP_BJOB 3 time windows		MRHP_BMCH- MSBH comparison				MRHP_FJOB - MSBH comparison				MRHP_BJOB - MSBH comparison			
										C_{max}		time		C_{max}		time		C_{max}		time	
		C_{max}	time	C_{max}	time	C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
10x5x[6.10]	avg	223.6	12.2	247.9	24.2	240.6	23.7	254.4	23.9	1.114	11.38	1.985	98.46	1.078	7.80	1.944	94.42	1.140	13.99	1.960	95.96
	min	177.0	12.0	225.0	23.0	205.0	23.0	219.0	23.0	1.017	1.70	1.846	84.62	0.945	-5.53	1.846	84.62	1.074	7.37	1.917	91.67
	max	241.0	13.0	297.0	26.0	267.0	24.0	284.0	25.0	1.311	31.07	2.083	108.33	1.186	18.64	2.000	100.00	1.237	23.73	2.000	100.00
10x10x[11.15]	avg	261.6	24.1	299.9	40.3	293.4	42.9	304.9	42.2	1.147	14.68	1.673	67.34	1.124	12.41	1.779	77.92	1.167	16.67	1.753	75.35
	min	227.0	23.0	259.0	38.0	257.0	38.0	267.0	38.0	1.050	5.00	1.560	56.00	1.046	4.56	1.625	62.50	1.088	8.78	1.520	52.00
	max	296.0	25.0	340.0	44.0	324.0	53.0	341.0	45.0	1.282	28.24	1.783	78.26	1.213	21.35	2.208	120.83	1.237	23.66	1.957	95.65
15x10x[11.15]	avg	346.1	36.8	363.5	55.4	380.6	55.5	382.3	62.4	1.054	5.42	1.509	50.88	1.103	10.27	1.511	51.06	1.108	10.81	1.703	70.27
	min	304.0	33.0	329.0	52.0	342.0	52.0	330.0	57.0	0.951	-4.89	1.366	36.59	1.000	0.00	1.366	36.59	0.997	-0.30	1.526	52.63
	max	413.0	41.0	413.0	59.0	424.0	59.0	435.0	68.0	1.197	19.74	1.636	63.64	1.211	21.05	1.583	58.33	1.233	23.29	2.061	106.06
15x15x[16.20]	avg	370.5	101.1	414.9	130.9	399.3	132.6	415.0	141.3	1.122	12.18	1.296	29.60	1.080	7.96	1.315	31.46	1.121	12.12	1.398	39.85
	min	350.0	96.0	388.0	117.0	368.0	125.0	366.0	131.0	1.012	1.24	1.182	18.18	0.976	-2.36	1.184	18.42	0.961	-3.94	1.323	32.32
	max	403.0	114.0	439.0	141.0	421.0	139.0	457.0	153.0	1.220	22.00	1.427	42.71	1.171	17.14	1.418	41.84	1.249	24.86	1.500	50.00
20x15x[16.20]	avg	439.5	171.4	478.2	208.3	469.1	202.6	504.1	221.4	1.089	8.91	1.217	21.69	1.069	6.88	1.184	18.39	1.148	14.85	1.291	29.09
	min	405.0	160.0	443.0	194.0	441.0	183.0	460.0	192.0	0.973	-2.67	1.102	10.23	0.996	-0.41	1.040	3.98	1.070	6.96	1.178	17.79
	max	487.0	183.0	544.0	224.0	501.0	216.0	561.0	267.0	1.180	18.00	1.282	28.22	1.136	13.65	1.269	26.88	1.302	30.16	1.517	51.70
20x20x[21.25]	avg	486.4	364.0	538.0	458.0	513.0	458.0	553.8	479.6	1.107	10.71	1.260	25.98	1.055	5.54	1.260	26.02	1.139	13.92	1.319	31.92
	min	470.0	330.0	499.0	427.0	464.0	430.0	522.0	450.0	1.000	0.00	1.158	15.80	0.974	-2.57	1.137	13.73	1.051	5.11	1.199	19.95
	max	509.0	386.0	590.0	481.0	556.0	496.0	584.0	521.0	1.219	21.90	1.352	35.24	1.175	17.55	1.355	35.53	1.222	22.20	1.416	41.58

Table 6.35 (Cont'd)

		MSBH		MRHP_BMCH 3 time windows		MRHP_FJOB 3 time windows		MRHP_BJOB 3 time windows		MRHP_BMCH- MSBH comparison				MRHP_FJOB - MSBH comparison				MRHP_BJOB - MSBH comparison			
										C_{max}		time		C_{max}		time		C_{max}		time	
		C_{max}	time	C_{max}	time	C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
30x15x[16.20]	avg	572.1	325.9	604.9	396.7	595.4	389.0	606.5	415.6	1.059	5.92	1.219	21.90	1.043	4.26	1.196	19.62	1.062	6.19	1.277	27.67
	min	528.0	313.0	544.0	363.0	531.0	362.0	546.0	400.0	0.913	-8.72	1.066	6.63	0.938	-6.21	1.025	2.49	0.970	-2.98	1.193	19.34
	max	638.0	362.0	650.0	431.0	663.0	429.0	673.0	432.0	1.117	11.74	1.311	31.11	1.169	16.94	1.371	37.06	1.160	16.03	1.324	32.38
30x20x[21.25]	avg	619.6	769.0	647.2	933.2	676.9	926.6	675.0	971.6	1.046	4.60	1.214	21.36	1.092	9.23	1.206	20.62	1.091	9.09	1.264	26.36
	min	571.0	747.0	609.0	864.0	605.0	836.0	626.0	931.0	0.949	-5.12	1.141	14.13	0.970	-3.04	1.099	9.87	1.028	2.78	1.203	20.30
	max	651.0	808.0	728.0	974.0	851.0	1390.0	745.0	1099.0	1.169	16.85	1.263	26.29	1.313	31.33	1.836	83.62	1.212	21.19	1.386	38.59
50x15x[16.20]	avg	827.3	840.6	864.3	918.2	900.5	917.3	901.1	1041.1	1.045	4.54	1.092	9.20	1.089	8.86	1.091	9.14	1.091	9.06	1.239	23.92
	min	770.0	774.0	774.0	850.0	818.0	843.0	833.0	943.0	0.989	-1.15	1.057	5.72	1.034	3.38	1.050	4.98	1.019	1.88	1.188	18.80
	max	960.0	887.0	949.0	1009.0	1042.0	981.0	978.0	1097.0	1.095	9.51	1.141	14.14	1.189	18.88	1.147	14.74	1.198	19.77	1.318	31.79
50x20x[21.25]	avg	865.6	1971.1	894.1	2140.2	901.9	2090.3	923.2	2339.6	1.037	3.71	1.086	8.62	1.047	4.67	1.061	6.07	1.071	7.11	1.187	18.72
	min	750.0	1861.0	825.0	1986.0	826.0	1989.0	873.0	2205.0	0.901	-9.87	1.030	3.02	0.926	-7.40	1.039	3.89	0.960	-4.03	1.120	12.00
	max	942.0	2025.0	1006.0	2335.0	1008.0	2176.0	988.0	2530.0	1.177	17.73	1.175	17.46	1.303	30.27	1.098	9.77	1.275	27.47	1.253	25.32
100x15x[16.20]	avg	1507.1	3359.9	1550.8	3208.9	1616.0	3239.5	1641.1	3759.0	1.029	2.92	0.956	-4.44	1.073	7.27	0.965	-3.48	1.090	8.97	1.120	12.02
	min	1390.0	3161.0	1437.0	2947.0	1458.0	3125.0	1503.0	3554.0	0.984	-1.60	0.880	-11.99	0.982	-1.79	0.899	-10.08	0.996	-0.38	1.007	0.69
	max	1580.0	3620.0	1645.0	3380.0	1707.0	3440.0	1731.0	3931.0	1.072	7.17	1.013	1.35	1.132	13.23	1.023	2.26	1.167	16.69	1.194	19.37
100x20x[21.25]	avg	1504.0	8188.1	1561.3	7381.7	1578.6	7320.1	1597.9	8301.7	1.039	3.85	0.902	-9.83	1.050	5.01	0.894	-10.58	1.063	6.32	1.014	1.45
	min	1435.0	7806.0	1488.0	7000.0	1527.0	7076.0	1506.0	7916.0	0.994	-0.64	0.823	-17.73	0.995	-0.50	0.864	-13.58	0.965	-3.49	0.950	-4.97
	max	1586.0	8509.0	1699.0	7976.0	1687.0	7676.0	1824.0	8707.0	1.123	12.29	0.947	-5.25	1.089	8.86	0.922	-7.84	1.171	17.15	1.093	9.30

Table 6.36 Comparison of the MRHP methods with three and two time windows

		MRHP_BMCH window3-window2 comparison				MRHP_FJOB window3-window2 comparison				MRHP_BJOB window3-window2 comparison			
		C_{max}		time		C_{max}		time		C_{max}		time	
		ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
10x5x[6.10]	avg	1.049	4.86	1.081	8.13	1.012	1.24	1.012	1.18	1.059	5.93	1.021	2.14
	min	0.911	-8.87	1.043	4.35	0.880	-12.02	0.857	-14.29	1.004	0.40	1.000	0.00
	max	1.172	17.17	1.143	14.29	1.123	12.30	1.043	4.35	1.164	16.39	1.043	4.35
10x10x[11.15]	avg	1.108	10.84	1.089	8.92	1.060	5.95	1.195	19.46	1.088	8.84	1.140	14.00
	min	0.974	-2.63	0.955	-4.55	0.923	-7.72	1.051	5.13	0.982	-1.83	1.083	8.33
	max	1.179	17.91	1.171	17.14	1.187	18.68	1.559	55.88	1.195	19.50	1.216	21.62
15x10x[11.15]	avg	1.028	2.77	1.012	1.17	1.048	4.79	1.101	10.12	1.037	3.75	1.176	17.55
	min	0.956	-4.37	0.831	-16.92	0.952	-4.84	1.077	7.69	0.944	-5.61	0.970	-3.03
	max	1.114	11.37	1.120	12.00	1.125	12.50	1.135	13.46	1.192	19.22	1.388	38.78
15x15x[16.20]	avg	1.024	2.43	1.064	6.42	1.026	2.59	1.174	17.44	1.050	4.99	1.127	12.73
	min	0.897	-10.28	0.936	-6.40	0.968	-3.16	1.098	9.76	0.969	-3.13	1.000	0.00
	max	1.134	13.39	1.151	15.13	1.139	13.86	1.252	25.23	1.111	11.14	1.214	21.43
20x15x[16.20]	avg	1.031	3.07	1.188	18.85	1.012	1.21	1.186	18.55	1.069	6.91	1.209	20.89
	min	0.966	-3.37	1.092	9.19	0.950	-4.99	1.118	11.76	1.020	2.00	1.103	10.34
	max	1.162	16.24	1.250	25.00	1.080	8.02	1.261	26.09	1.175	17.46	1.315	31.53
20x20x[21.25]	avg	1.082	8.17	1.153	15.34	1.060	6.05	1.227	22.72	1.078	7.80	1.204	20.40
	min	0.968	-3.21	1.032	3.24	0.962	-3.78	1.167	16.71	0.987	-1.29	1.125	12.50
	max	1.216	21.65	1.206	20.62	1.153	15.30	1.329	32.86	1.136	13.61	1.310	31.03
30x15x[16.20]	avg	1.031	3.09	1.184	18.40	1.034	3.42	1.207	20.68	1.019	1.86	1.210	20.96
	min	0.953	-4.71	0.926	-7.43	0.958	-4.15	0.971	-2.88	0.942	-5.78	1.096	9.57
	max	1.145	14.51	1.283	28.26	1.144	14.44	1.336	33.64	1.105	10.53	1.271	27.13
30x20x[21.25]	avg	1.020	2.04	1.162	16.21	1.069	6.93	1.273	27.32	1.043	4.30	1.233	23.27
	min	0.949	-5.12	0.993	-0.69	0.988	-1.24	1.142	14.17	0.994	-0.61	1.165	16.55
	max	1.075	7.48	1.285	28.51	1.359	35.94	1.941	94.13	1.117	11.69	1.342	34.19
50x15x[16.20]	avg	1.011	1.13	1.187	18.66	1.058	5.79	1.138	13.78	1.045	4.47	1.239	23.93
	min	0.945	-5.49	1.139	13.94	1.024	2.43	0.937	-6.30	0.984	-1.63	1.180	18.02
	max	1.078	7.78	1.226	22.56	1.113	11.35	1.213	21.26	1.121	12.06	1.291	29.12
50x20x[21.25]	avg	1.040	4.03	1.195	19.47	1.039	3.85	1.182	18.20	1.049	4.89	1.220	22.02
	min	1.005	0.47	1.066	6.57	0.978	-2.25	1.136	13.63	1.008	0.79	1.172	17.22
	max	1.114	11.41	1.327	32.73	1.122	12.17	1.219	21.93	1.138	13.81	1.281	28.07
100x15x[16.20]	avg	1.003	0.26	1.136	13.57	1.023	2.31	1.144	14.45	1.036	3.59	1.218	21.77
	min	0.957	-4.32	1.065	6.50	0.996	-0.36	1.108	10.79	0.953	-4.67	1.174	17.39
	max	1.039	3.92	1.211	21.05	1.055	5.50	1.179	17.93	1.093	9.31	1.252	25.19
100x20x[21.25]	avg	1.024	2.44	1.121	12.14	1.020	2.00	1.100	9.97	1.034	3.35	1.166	16.64
	min	0.974	-2.60	1.002	0.16	0.971	-2.85	1.060	6.05	0.977	-2.25	1.029	2.89
	max	1.097	9.73	1.170	17.00	1.101	10.09	1.140	14.02	1.101	10.11	1.206	20.57

In the first experiment of this subsection, we examine the performance of the MSBH methods on the problems. The summary of the average results are shown in Table 6.37. As it is seen from Table 6.37, in average, the MSBH is slightly outperformed by the MSBH_TML and the MSBH_TOTINF based on solution quality. Furthermore, the MSBH_TOTINF outperforms the MSBH_TML based on solution quality. The MSBH outperforms the MSBH_TOTINF, and is outperformed by the MSBH_TML based on solution time. It is interesting to note that, especially for large problems, the MSBH_TOTINF spends too much time to solve the problem. Because, calculating infeasibility profiles take too much time.

There is no significant difference in average solution quality among the three MSBH methods, although there are sometimes significant differences for given instances. The MSBH_TML and the MSBH_TOTINF slightly outperform the MSBH on solution quality in most of the job-machine combinations. The MSBH_TML needs less computational time, whereas the MSBH_TOTINF finds the solutions in more seconds. In general, the MSBH_TML obtain solutions almost as good as the MSBH_TOTINF in about 141.95% less computational times.

The degree of difficulty of solving a problem by the MSBH of course sharply increases with the number of machines. However, for a given number of machines, an increase in the number of jobs does not seem to make the problem more difficult; on the contrary, an increase in the number of machines while the number of jobs is constant increase computational times much more. For example, as it is seen in Table 6.37, for the MSBH, the average solution time is 171.4 seconds for 20 jobs - 15 machines instances, whereas it is 325.9 seconds for 30 jobs - 15 machines instances, and it is 364 seconds for 20 jobs - 20 machines problems. As it is seen, if we increase the number of machines, the solution time increases more than we increase the number of jobs.

Table 6.37 Summary of the results for the MSBH methods

	MSBH		MSBH_TML		MSBH_TOTINF		MSBH_TML - MSBH comparison				MSBH_TOTINF - MSBH comparison				MSBH_TOTINF - MSBH_TML comparison			
	C_{max}	time	C_{max}	time	C_{max}	time	C_{max}		time		C_{max}		time		C_{max}		time	
							ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
10x5x[6.10]	223.6	12.2	225.8	11.7	226.4	15.4	1.011	1.06	0.960	-4.04	1.013	1.32	1.262	26.22	1.003	0.30	1.317	31.67
10x10x[11.15]	261.6	24.1	266.7	22.4	255.9	44.4	1.022	2.24	0.930	-6.99	0.981	-1.90	1.844	84.44	0.960	-3.96	1.985	98.45
15x10x[11.15]	346.1	36.8	343.0	32.6	338.4	75.5	0.993	-0.74	0.887	-11.30	0.980	-2.00	2.050	105.00	0.988	-1.21	2.313	131.32
15x15x[16.20]	370.5	101.1	365.2	88.1	368.8	225.6	0.988	-1.24	0.873	-12.74	0.997	-0.29	2.237	123.71	1.011	1.06	2.563	156.30
20x15x[16.20]	439.5	171.4	444.9	136.7	433.9	355.0	1.015	1.50	0.798	-20.23	0.989	-1.13	2.071	107.11	0.977	-2.34	2.602	160.25
20x20x[21.25]	486.4	364.0	480.3	296.8	470.8	802.2	0.988	-1.20	0.816	-18.42	0.968	-3.17	2.205	120.51	0.982	-1.80	2.703	170.34
30x15x[16.20]	572.1	325.9	558.6	255.7	559.5	700.5	0.977	-2.29	0.785	-21.46	0.980	-2.05	2.149	114.93	1.003	0.32	2.739	173.90
30x20x[21.25]	619.6	769.0	616.0	613.0	623.0	1599.3	0.996	-0.37	0.797	-20.29	1.007	0.69	2.080	107.96	1.012	1.21	2.610	160.98
50x15x[16.20]	827.3	840.6	832.0	650.9	842.4	1760.5	1.006	0.59	0.775	-22.54	1.016	1.63	2.091	109.09	1.011	1.09	2.700	170.04
50x20x[21.25]	865.6	1971.1	857.4	1515.1	842.3	3840.0	0.993	-0.72	0.769	-23.15	0.976	-2.40	1.948	94.80	0.983	-1.68	2.536	153.58
100x15x[16.20]	1507.1	3359.9	1492.8	2488.5	1491.6	6345.3	0.990	-0.95	0.741	-25.90	0.990	-1.03	1.890	88.96	0.999	-0.07	2.550	154.98
100x20x[21.25]	1504.0	8188.1	1461.5	5893.3	1460.7	14234.3	0.972	-2.78	0.720	-28.03	0.972	-2.84	1.739	73.86	1.000	-0.05	2.416	141.58
avg.	**668.6**	**1347.0**	**662.0**	**1000.4**	**659.5**	**2499.8**	**0.996**	**-0.41**	**0.821**	**-17.93**	**0.989**	**-1.10**	**1.964**	**96.38**	**0.994**	**-0.59**	**2.420**	**141.95**

The average results obtained by the MSBH were compared with the results obtained by the MRHP in Tables 6.38 and 6.39. Furthermore, Table 6.40 represents the comparison of the average results for the MRHP with two and three time windows. We conclude that the MSBH outperforms all the MRHP methods on solution quality. The MSBH finds consistently better solutions of almost all the instances than the MRHP does. In average, the MSBH is outperformed by the MRHP with two and three time windows based on solution time. However, the MSBH outperforms the MRHP_BJOB with three time windows based on time. Less than 50 jobs, the MSBH solves the problems in less times. The average times of the MSBH is shorter than the MRHP with two time windows for 10, 20, and 30 jobs. Furthermore, after 50 jobs, the MRHP with two time windows begins to solve instances less computational times than the MSBH. For example, for 50 jobs - 20 machines instances, solution times obtained by the MSBH are 1971.1 seconds on average, whereas it is 1769.3 seconds for the MRHP_FJOB with two time windows. Especially for large size instances with more than 50 jobs, it is clear that the MRHP is much faster than the other methods.

As it is expected, when the number of time windows increase, the quality of solution is decreased. The solution quality of the MRHP with two time windows is better than three time windows. From Table 6.40, it is clear that in terms of quality, the solution for two window is better than the solution for three time windows. This holds true since the decomposition process will always make mistakes on assigning operations to each time window. An interesting result for the MRHP is that the solution times increase when the number of time windows are increased. Average solution times of two time windows are less than three time windows. This is due to the sequence based algorithm of the MSBH which is used in the MRHP.

Table 6.38 Summary of the results for the MRHP with two time windows

	MSBH		MRHP_BMCH 2 time windows		MRHP_FJOB 2 time windows		MRHP_BJOB 2 time windows		MRHP_BMCH- MSBH comparison				MRHP_FJOB - MSBH comparison				MRHP_BJOB - MSBH comparison			
	C_{max}	time							C_{max}		time		C_{max}		time		C_{max}		time	
			C_{max}	time	C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
10x5x[6.10]	223.6	12.2	238.0	22.4	238.3	23.5	240.2	23.4	1.064	6.37	1.837	83.72	1.066	6.56	1.926	92.56	1.078	7.78	1.919	91.92
10x10x[11.15]	261.6	24.1	270.6	37.1	278.0	36.0	280.7	37.0	1.037	3.67	1.540	53.97	1.063	6.33	1.495	49.50	1.075	7.53	1.537	53.69
15x10x[11.15]	346.1	36.8	354.3	55.1	364.3	50.4	370.0	53.5	1.025	2.51	1.505	-83.90	1.053	5.33	1.372	37.19	1.069	6.92	1.452	45.19
15x15x[16.20]	370.5	101.1	406.7	123.1	389.7	113.0	395.4	125.5	1.099	9.92	1.220	21.97	1.054	5.37	1.119	11.91	1.069	6.86	1.244	24.44
20x15x[16.20]	439.5	171.4	464.8	175.5	464.0	171.1	472.3	183.0	1.059	5.87	1.025	2.45	1.058	5.83	0.998	-0.15	1.075	7.49	1.067	6.75
20x20x[21.25]	486.4	364.0	498.1	397.9	484.1	373.3	514.5	399.0	1.025	2.46	1.093	9.35	0.996	-0.40	1.027	2.73	1.058	5.83	1.098	9.77
30x15x[16.20]	572.1	325.9	587.9	337.6	575.9	323.6	595.8	344.1	1.030	3.00	1.034	3.36	1.009	0.88	0.992	-0.76	1.04	4.278	1.06	5.7
30x20x[21.25]	619.6	769.0	634.4	808.1	633.8	728.7	647.2	788.4	1.025	2.54	1.050	5.02	1.024	2.38	0.948	-5.23	1.046	4.58	1.025	2.53
50x15x[16.20]	827.3	840.6	855.0	773.7	850.7	811.0	863.2	840.3	1.034	3.44	0.921	-7.94	1.029	2.91	0.963	-3.68	1.045	4.49	1.000	0.01
50x20x[21.25]	865.6	1971.1	859.3	1795.2	868.0	1769.3	880.6	1917.5	0.997	-0.35	0.911	-8.94	1.006	0.59	0.898	-10.23	1.021	2.05	0.973	-2.70
100x15x[16.20]	1507.1	3359.9	1547.3	2829.0	1579.8	2830.3	1585.5	3087.2	1.027	2.69	0.842	-15.79	1.048	4.84	0.843	-15.68	1.052	5.24	0.920	-8.03
100x20x[21.25]	1504.0	8188.1	1524.9	6588.8	1548.5	6658.0	1545.2	7126.5	1.014	1.44	0.804	-19.55	1.030	3.04	0.813	-18.66	1.028	2.82	0.870	-12.95
avg.	**668.6**	**1347.0**	**686.8**	**1162.0**	**689.6**	**1157.4**	**699.2**	**1243.8**	**1.036**	**3.63**	**1.148**	**3.64**	**1.036**	**3.64**	**1.116**	**11.62**	**1.055**	**5.49**	**1.180**	**18.03**

Table 6.39 Summary of the results for the MRHP with three time windows

	MSBH		MRHP_BMCH 3 time windows		MRHP_FJOB 3 time windows		MRHP_BJOB 3 time windows		MRHP_BMCH- MSBH comparison				MRHP_FJOB - MSBH comparison				MRHP_BJOB - MSBH comparison			
	C_{max}	time							C_{max}		time		C_{max}		time		C_{max}		time	
			C_{max}	time	C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
10x5x[6.10]	223.6	12.2	247.9	24.2	240.6	23.7	254.4	23.9	1.114	11.38	1.985	98.46	1.078	7.80	1.944	94.42	1.140	13.99	1.960	95.96
10x10x[11.15]	261.6	24.1	299.9	40.3	293.4	42.9	304.9	42.2	1.147	14.68	1.673	67.34	1.124	12.41	1.779	77.92	1.167	16.67	1.753	75.35
15x10x[11.15]	346.1	36.8	363.5	55.4	380.6	55.5	382.3	62.4	1.054	5.42	1.509	50.88	1.103	10.27	1.511	51.06	1.108	10.81	1.703	70.27
15x15x[16.20]	370.5	101.1	414.9	130.9	399.3	132.6	415.0	141.3	1.122	12.18	1.296	29.60	1.080	7.96	1.315	31.46	1.121	12.12	1.398	39.85
20x15x[16.20]	439.5	171.4	478.2	208.3	469.1	202.6	504.1	221.4	1.089	8.91	1.217	21.69	1.069	6.88	1.184	18.39	1.148	14.85	1.291	29.09
20x20x[21.25]	486.4	364.0	538.0	458.0	513.0	458.0	553.8	479.6	1.107	10.71	1.260	25.98	1.055	5.54	1.260	26.02	1.139	13.92	1.319	31.92
30x15x[16.20]	572.1	325.9	604.9	396.7	595.4	389.0	606.5	415.6	1.059	5.92	1.219	21.90	1.043	4.26	1.196	19.62	1.062	6.19	1.277	27.67
30x20x[21.25]	619.6	769.0	647.2	933.2	676.9	926.6	675.0	971.6	1.046	4.60	1.214	21.36	1.092	9.23	1.206	20.62	1.091	9.09	1.264	26.36
50x15x[16.20]	827.3	840.6	864.3	918.2	900.5	917.3	901.1	1041.1	1.045	4.54	1.092	9.20	1.089	8.86	1.091	9.14	1.091	9.06	1.239	23.92
50x20x[21.25]	865.6	1971.1	894.1	2140.2	901.9	2090.3	923.2	2339.6	1.037	3.71	1.086	8.62	1.047	4.67	1.061	6.07	1.071	7.11	1.187	18.72
100x15x[16.20]	1507.1	3359.9	1550.8	3208.9	1616.0	3239.5	1641.1	3759.0	1.029	2.92	0.956	-4.44	1.073	7.27	0.965	-3.48	1.090	8.97	1.120	12.02
100x20x[21.25]	1504.0	8188.1	1561.3	7381.7	1578.6	7320.1	1597.9	8301.7	1.039	3.85	0.902	-9.83	1.050	5.01	0.894	-10.58	1.063	6.32	1.014	1.45
avg.	**668.6**	**1347.0**	**705.4**	**1324.7**	**713.8**	**1316.5**	**729.9**	**1483.3**	**1.074**	**7.40**	**1.284**	**28.40**	**1.075**	**7.51**	**1.284**	**28.39**	**1.108**	**10.76**	**1.377**	**37.71**

Table 6.40 Summary of the results for the MRHP with two and three time windows

	MRHP_BMCH window3-window2 comparison				MRHP_FJOB window3-window2 comparison				MRHP_BJOB window3-window2 comparison			
	C_{max}		time		C_{max}		time		C_{max}		time	
	ratio	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**	**ratio**	**RE(%)**
10x5x[6.10]	1.049	4.86	1.081	8.13	1.012	1.24	1.012	1.18	1.059	5.93	1.021	2.14
10x10x[11.15]	1.108	10.84	1.089	8.92	1.060	5.95	1.195	19.46	1.088	8.84	1.140	14.00
15x10x[11.15]	1.028	2.77	1.012	1.17	1.048	4.79	1.101	10.12	1.037	3.75	1.176	17.55
15x15x[16.20]	1.024	2.43	1.064	6.42	1.026	2.59	1.174	17.44	1.050	4.99	1.127	12.73
20x15x[16.20]	1.031	3.07	1.188	18.85	1.012	1.21	1.186	18.55	1.069	6.91	1.209	20.89
20x20x[21.25]	1.082	8.17	1.153	15.34	1.060	6.05	1.227	22.72	1.078	7.80	1.204	20.40
30x15x[16.20]	1.031	3.09	1.184	18.40	1.034	3.42	1.207	20.68	1.019	1.86	1.210	20.96
30x20x[21.25]	1.020	2.04	1.162	16.21	1.069	6.93	1.273	27.32	1.043	4.30	1.233	23.27
50x15x[16.20]	1.011	1.13	1.187	18.66	1.058	5.79	1.138	13.78	1.045	4.47	1.239	23.93
50x20x[21.25]	1.040	4.03	1.195	19.47	1.039	3.85	1.182	18.20	1.049	4.89	1.220	22.02
100x15x[16.20]	1.003	0.26	1.136	13.57	1.023	2.31	1.144	14.45	1.036	3.59	1.218	21.77
100x20x[21.25]	1.024	2.44	1.121	12.14	1.020	2.00	1.100	9.97	1.034	3.35	1.166	16.64
avg.	**1.04**	**3.76**	**1.13**	**13.11**	**1.04**	**3.84**	**1.16**	**16.16**	**1.05**	**5.06**	**1.18**	**18.02**

Finally, we compared the results to evaluate the differences using the paired t-test at the 1% level of significance. Table 6.41 presents summary of all the results. In Table 6.41, the results whose difference is not statistically significant are highlighted in bold type. The difference of the results of the MSBH_TML and the MSBH_TOTINF with the MSBH are not statistically significant based on solution quality, whereas it is significant based on solution time. Furthermore, the difference of the results of the MSBH_TML and the MSBH_TOTINF is not statistically significant based on solution quality, whereas it is significant on solution time. Further analysis of Table 6.41 reveals that the difference of the results of the MRHP methods for two and three time windows are statistically significant on both solution quality and time.

Table 6.41 The summary of the results for large problems

	C_{max}	time
MSBH_TML - MSBH	**0.996**	0.821
MSBH_TOTINF - MSBH	**0.989**	1.964
MSBH_TOTINF - MSBH_TML	**0.994**	2.420
BMCH(window2) - MSBH	1.036	1.148
BMCH(window3) - MSBH	1.074	1.284
FJOB(window2) - MSBH	1.036	1.116
FJOB(window3) - MSBH	1.075	1.284
BJOB(window2) - MSBH	1.055	1.180
BJOB(window3) - MSBH	1.108	1.377
BMCH(window3) - BMCH(window2)	1.038	1.131
FJOB(window3) - FJOB(window2)	1.038	1.162
BJOB(window3) - BJOB(window2)	1.051	1.180

In summary, our results on the large C_{max} problems yield a number of results. The solution times for the MSBH methods increase for large problems, nevertheless it can solve large size problems in real world. The MSBH methods consistently outperform the MRHP in terms of solution quality.

In Table 6.42, we compared the results to evaluate the differences using the paired t-test at the 1% level of significance using the small and large size problems together. Analysis of the table reveals that the differences of results of the MSBH, the MSBH_TML, and the MSBH_TOTINF are not statistically significant.

Table 6.42 The summary small and large size problems

	C_{max}	time
MSBH_TML - MSBH	**0.998**	0.867
MSBH_TOTINF - MSBH	**0.996**	1.602
MSBH_TOTINF - TML	**0.998**	1.897
FJOB(2window) - MSBH	1.043	1.590
FJOB(3window) - MSBH	1.081	1.639
BMCH(2window) - MSBH	1.049	1.604
BMCH(3window) - MSBH	1.087	1.643
BJOB(2window) - MSBH	1.062	1.416
BJOB(3window) - MSBH	1.101	1.680
FJOB(3window) - FJOB(2window)	1.037	1.049
BMCH(3window) - BMCH(2window)	1.038	1.065
BJOB(3window) - BJOB(2window)	1.038	1.068

6.5 Comparison Of The Proposed Methods For The Special Reentrant Job Shop Scheduling Problem

The last set of test problems is generated for the special RJSSP which is the application problem of this study. These instances are designed to represent the characteristics of the textile facility. In this reentrant job shop, each job goes for its processing first to a machine called the primary machine, then to a number of other machines, and finally to a machine for its last operation. Furthermore, these two machines are not used anywhere of the job shop additionally. For example, if there are 10 machines in the shop, for all jobs, the first machine is machine 1, the last machine is machine 10. These problems are solved by the MRHP_BMCH, the MRHP_FJOB and the MRHP_BJOB rules for assignment of operations to time windows for the classical RJSSP which are described in Section 5.1.

Furthermore, the special rule (MRHP_SR) which was improved for the application problem in Section 5.2 is used to solve the problems. Different sizes of problems are used to see the effect of the rules on solution time and quality. In addition, different number of time windows are tested to see the influence of the number of time windows on the objective function value and solution times.

The problems are generated in MATLAB with additional constraints to the test problem generation procedure. To see the effect of this special RJSSP on makespan, like in the application problem, the processing times of these machines are generated longer than those of the other machines. The processing times are randomly drawn from a uniform probability distribution on the interval [25,75] for the first machine, [50,100] for the last machine, and [1,50] for the other machines.

In the experiments carried out, the MSBH_TML which is used in the MRHP was implemented with one resequencing cycle. Applying more than one is time consuming, especially if it is applied to the job shops involving too many jobs.

The problems are randomly generated by setting the number of jobs at one of the four levels: 10, 20, 30, 50,100; and the number of machines at one of the three levels: 10, 15, 20. The job-machine configurations are as follows:

- 10x10x[11,15]
- 10x15x[16,20]
- 10x20x[21,25]
- 20x10x[11,15]
- 20x15x[16,20]
- 20x20x[21,25]
- 30x10x[11,15]
- 30x15x[16,20]
- 30x20x[21,25]
- 50x10x[11,15]
- 50x20x[21,25]
- 100x10x[11,15]

Ten instances per configuration are generated yielding 120 problems overall. The number of time windows are set to two, three, and five according to the problem size.

6.5.1 Details Of The Implementation And The Computational Results

The complete results for all test problems obtained by alternative rules for assignment of operations to time windows are given in Appendix B. The number of time windows that are considered in the MRHP for the 10 job instances are two and three, for the 20, 30, 50 and 100 jobs instances are two, three and five. The results of this experiment are summarized in Table 6.42. The results show the ratios of the C_{max} and time values of the MRHP_BMCH, the MHRP_FJOB and the MRHP_BJOB to the MRHP_SR. Summary of the results for two, three and five time windows are shown in Table 6.43.

Table 6.42 The results of the MRHP methods for 10 jobs instances

			MRHP_SR		MRHP_BMCH		BMCH - SR comparison				MRHP_FJOB		FJOB - SR comparison				MRHP_BJOB		BJOB - SR comparison			
							C_{max}		time				C_{max}		time				C_{max}		time	
	w		C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)
10x10x[11.15]	2	avg	1248.7	36.4	1389.7	36.1	1.119	11.86	1.000	-0.01	1480.2	34.4	1.191	19.15	0.952	-4.82	1453.1	35.3	1.171	17.11	0.977	-2.34
		min	1559.0	45.0	1646.0	42.0	1.225	22.47	1.200	20.00	1736.0	37.0	1.322	32.22	1.057	5.71	1575.0	38.0	1.325	32.47	1.086	8.57
		max	1133.0	34.0	1228.0	34.0	0.935	-6.47	0.756	-24.44	1361.0	33.0	1.025	2.54	0.733	-26.67	1297.0	34.0	1.010	1.03	0.756	-24.44
	3	avg	1343.8	37.6	1405.1	39.6	1.054	5.43	1.055	5.48	1550.7	37.8	1.163	16.33	1.007	0.67	1496.2	37.8	1.122	12.17	1.009	0.85
		min	1660.0	43.0	1583.0	50.0	1.254	25.37	1.316	31.58	1692.0	45.0	1.353	35.31	1.118	11.76	1643.0	44.0	1.254	25.37	1.257	25.71
		max	1127.0	34.0	1253.0	35.0	0.867	-13.35	0.907	-9.30	1465.0	35.0	1.019	1.93	0.921	-7.89	1369.0	33.0	0.990	-1.02	0.892	-10.81
10x15x[16.20]	2	avg	1432.7	72.0	1521.0	77.1	1.062	6.18	1.077	7.69	1581.5	68.0	1.105	10.53	0.948	-5.19	1590.0	69.0	1.111	11.15	0.963	-3.75
		min	1506.0	82.0	1641.0	95.0	1.127	12.71	1.357	35.71	1699.0	73.0	1.188	18.84	1.092	9.23	1756.0	75.0	1.206	20.60	1.108	10.77
		max	1316.0	65.0	1339.0	69.0	0.990	-1.05	0.950	-5.00	1495.0	64.0	1.053	5.27	0.829	-17.07	1496.0	65.0	0.993	-0.66	0.805	-19.51
	3	avg	1650.7	76.9	1615.6	83.7	0.981	-1.94	1.089	8.86	1685.1	77.7	1.022	2.24	1.012	1.16	1680.7	78.1	1.020	2.00	1.017	1.70
		min	1728.0	83.0	1832.0	90.0	1.087	8.72	1.147	14.67	1869.0	82.0	1.092	9.17	1.079	7.89	1813.0	91.0	1.087	8.69	1.182	18.18
		max	1462.0	74.0	1490.0	78.0	0.872	-12.76	1.013	1.30	1546.0	74.0	0.940	-6.02	0.904	-9.64	1496.0	71.0	0.866	-13.43	0.892	-10.84
10x20x[21.25]	2	avg	1663.1	141.3	1639.7	155.0	0.986	-1.38	1.098	9.84	1728.8	151.7	1.040	3.99	1.075	7.46	1692.2	159.2	1.017	1.74	1.127	12.70
		min	1785.0	152.0	1828.0	165.0	1.090	8.98	1.170	17.02	1892.0	172.0	1.138	13.77	1.255	25.55	1869.0	173.0	1.069	6.92	1.241	24.09
		max	1558.0	137.0	1499.0	141.0	0.931	-6.89	0.928	-7.24	1543.0	137.0	0.965	-3.53	0.958	-4.20	1556.0	149.0	0.958	-4.20	1.042	4.20
	3	avg	1937.3	149.7	1839.6	167.8	0.950	-5.01	1.123	12.29	1740.4	177.0	0.899	-10.10	1.186	18.62	1739.3	160.4	0.898	-10.18	1.075	7.51
		min	2060.0	176.0	2040.0	191.0	1.053	5.32	1.223	22.30	1892.0	210.0	0.977	-2.32	1.400	40.00	1869.0	186.0	0.958	-4.23	1.265	26.53
		max	1882.0	138.0	1578.0	150.0	0.838	-16.15	1.000	0.00	1591.0	159.0	0.815	-18.50	0.949	-5.11	1591.0	139.0	0.814	-18.64	0.903	-9.66

Table 6.42 (Cont'd)

			MRHP_SR		MRHP_BMCH		BMCH - SR comparison				MRHP_FJOB		FJOB - SR comparison				MRHP_BJOB		BJOB - SR comparison			
							C_{max}		time				C_{max}		time				C_{max}		time	
	w		C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)
20x10x[11.15]	2	avg	2045.2	67.1	2556.2	77.3	1.251	25.08	1.153	15.35	2763.8	67.5	1.352	35.19	1.007	0.66	2702.7	75.4	1.324	32.36	1.124	12.41
		min	2225.0	73.0	2747.0	83.0	1.343	34.33	1.266	26.56	3038.0	74.0	1.408	40.81	1.047	4.69	2851.0	82.0	1.423	42.32	1.206	20.59
		max	1921.0	64.0	2352.0	71.0	1.150	15.01	1.060	5.97	2618.0	64.0	1.271	27.09	0.928	-7.25	2615.0	69.0	1.244	24.45	1.045	4.48
	3	avg	2031.0	77.8	2534.2	81.6	1.250	25.02	1.051	5.13	2858.1	78.3	1.410	40.95	1.008	0.83	2793.6	81.2	1.377	37.66	1.044	4.38
		min	2217.0	91.0	2672.0	87.0	1.375	37.48	1.167	16.67	2991.0	85.0	1.508	50.84	1.083	8.33	3269.0	98.0	1.501	50.08	1.156	15.58
		max	1851.0	72.0	2341.0	75.0	1.171	17.13	0.956	-4.40	2739.0	74.0	1.349	34.91	0.934	-6.59	2584.0	69.0	1.285	28.48	0.896	-10.39
	5	avg	2242.4	99.0	2596.9	97.4	1.159	15.95	0.986	-1.41	3021.8	101.2	1.349	34.87	1.024	2.38	3037.1	99.2	1.356	35.56	1.003	0.28
		min	2376.0	111.0	2708.0	104.0	1.251	25.14	1.079	7.87	3291.0	111.0	1.443	44.28	1.124	12.36	3393.0	111.0	1.488	48.75	1.121	12.12
		max	2104.0	89.0	2433.0	83.0	1.048	4.83	0.838	-16.16	2866.0	94.0	1.257	25.67	0.949	-5.05	2855.0	89.0	1.244	24.43	0.919	-8.08
20x15x[16.20]	2	avg	2173.8	177.7	2598.2	193.1	1.197	19.67	1.089	8.90	2872.3	184.0	1.323	32.31	1.038	3.79	2837.3	199.1	1.307	30.71	1.122	12.24
		min	2377.0	196.0	2764.0	207.0	1.304	30.36	1.169	16.95	3030.0	202.0	1.475	47.47	1.195	19.53	2941.0	221.0	1.412	41.24	1.254	25.44
		max	1993.0	167.0	2402.0	170.0	1.118	11.78	0.944	-5.56	2687.0	171.0	1.275	27.47	0.950	-5.00	2613.0	178.0	1.194	19.35	1.011	1.11
	3	avg	2264.4	211.2	2706.1	220.0	1.197	19.66	1.044	4.40	3091.9	210.2	1.367	36.70	1.000	0.01	2905.5	226.0	1.285	28.46	1.075	7.55
		min	2385.0	234.0	2957.0	242.0	1.299	29.92	1.128	12.82	3186.0	222.0	1.485	48.55	1.168	16.76	3093.0	264.0	1.382	38.20	1.257	25.71
		max	2136.0	185.0	2468.0	196.0	1.057	5.70	0.936	-6.39	2989.0	190.0	1.304	30.36	0.889	-11.11	2733.0	196.0	1.200	20.03	0.925	-7.55
	5	avg	2672.5	278.2	2894.2	256.2	1.084	8.44	0.928	-7.22	3176.3	301.4	1.190	19.04	1.091	9.13	3121.8	266.3	1.170	16.96	0.965	-3.53
		min	2911.0	342.0	3255.0	288.0	1.196	19.63	1.108	10.77	3329.0	332.0	1.283	28.25	1.286	28.57	3309.0	309.0	1.234	23.39	1.188	18.85
		max	2481.0	245.0	2582.0	218.0	0.925	-7.52	0.765	-23.51	3036.0	264.0	1.117	11.68	0.930	-7.02	2998.0	233.0	1.085	8.45	0.822	-17.84
20x20x[21.25]	2	avg	2429.5	409.2	2672.1	424.7	1.101	10.07	1.039	3.92	2963.0	407.8	1.220	22.00	0.997	-0.25	2939.8	439.6	1.210	21.04	1.075	7.50
		min	2581.0	431.0	2776.0	462.0	1.184	18.38	1.109	10.88	3157.0	472.0	1.337	33.73	1.095	9.51	3053.0	483.0	1.247	24.75	1.148	14.83
		max	2326.0	376.0	2481.0	406.0	1.058	5.81	0.972	-2.84	2741.0	373.0	1.163	16.34	0.927	-7.35	2794.0	396.0	1.180	18.02	0.938	-6.16
	3	avg	2643.1	483.1	2895.4	508.6	1.097	9.67	1.052	5.23	3112.1	506.6	1.179	17.89	1.049	4.88	3099.3	531.0	1.175	17.47	1.100	9.96
		min	2886.0	504.0	3100.0	576.0	1.223	22.29	1.143	14.29	3348.0	530.0	1.236	23.63	1.100	9.96	3285.0	572.0	1.255	25.51	1.168	16.77
		max	2442.0	468.0	2692.0	474.0	1.035	3.49	1.006	0.64	2933.0	476.0	1.117	11.74	0.976	-2.40	2912.0	476.0	1.073	7.29	0.954	-4.61
	5	avg	3093.7	626.5	3230.3	619.9	1.045	4.55	0.991	-0.93	3247.7	687.9	1.051	5.14	1.098	9.83	3143.7	622.3	1.018	1.79	0.994	-0.59
		min	3282.0	686.0	3587.0	677.0	1.160	15.97	1.081	8.15	3498.0	774.0	1.123	12.30	1.166	16.57	3341.0	714.0	1.085	8.50	1.101	10.12
		max	2812.0	563.0	2899.0	563.0	0.964	-3.58	0.925	-7.53	3084.0	635.0	0.940	-6.03	0.995	-0.47	3003.0	577.0	0.926	-7.40	0.922	-7.83

Table 6.42 (Cont'd)

			MRHP_SR		MRHP_BMCH		BMCH - SR comparison				MRHP_FJOB		FJOB - SR comparison				MRHP_BJOB		BJOB - SR comparison			
							C_{max}		time				C_{max}		time				C_{max}		time	
	w		C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)
30x10x[11.15]	2	avg	2721.4	120.7	3689.1	134.5	1.357	35.65	1.114	11.44	4082.8	119.6	1.503	50.27	0.992	-0.84	3874.3	148.0	1.426	42.63	1.226	22.64
		min	2901.0	130.0	4047.0	150.0	1.441	44.06	1.172	17.24	4318.0	128.0	1.666	66.59	1.025	2.52	4032.0	180.0	1.545	54.48	1.513	51.26
		max	2592.0	116.0	3457.0	123.0	1.285	28.51	1.042	4.24	3934.0	114.0	1.388	38.83	0.931	-6.92	3490.0	125.0	1.284	28.36	1.059	5.93
	3	avg	2783.7	139.9	3749.5	153.8	1.348	34.79	1.099	9.91	4222.8	155.7	1.518	51.84	1.115	11.46	4073.0	152.7	1.465	46.49	1.092	9.19
		min	2915.0	150.0	3935.0	179.0	1.416	41.62	1.193	19.33	4395.0	179.0	1.590	59.04	1.336	33.58	4379.0	169.0	1.617	61.71	1.164	16.42
		max	2588.0	130.0	3360.0	147.0	1.161	16.14	1.028	2.80	4076.0	139.0	1.416	41.58	1.007	0.72	3898.0	144.0	1.381	38.11	1.041	4.11
	5	avg	3012.8	181.0	3875.6	178.4	1.290	29.03	0.987	-1.26	4483.7	197.9	1.493	49.34	1.093	9.33	4402.2	183.7	1.467	46.65	1.016	1.61
		min	3464.0	197.0	4035.0	199.0	1.368	36.83	1.131	13.07	4749.0	229.0	1.624	62.41	1.218	21.81	4697.0	198.0	1.565	56.48	1.119	11.86
		max	2830.0	168.0	3655.0	161.0	1.165	16.48	0.890	-11.05	4292.0	179.0	1.336	33.57	0.989	-1.09	4247.0	169.0	1.271	27.08	0.956	-4.37
30x15x[16.20]	2	avg	2930.5	353.0	3690.1	387.6	1.260	26.03	1.099	9.94	4064.1	355.4	1.388	38.80	1.009	0.87	4036.4	407.8	1.379	37.87	1.157	15.71
		min	3090.0	381.0	3795.0	448.0	1.322	32.23	1.269	26.91	4161.0	375.0	1.432	43.20	1.072	7.16	4148.0	429.0	1.456	45.64	1.245	24.48
		max	2788.0	328.0	3582.0	359.0	1.202	20.23	1.000	0.00	3920.0	337.0	1.315	31.52	0.931	-6.91	3876.0	380.0	1.304	30.44	1.051	5.15
	3	avg	3010.3	406.0	3843.0	432.3	1.277	27.69	1.065	6.48	4427.0	442.1	1.472	47.24	1.090	8.97	4235.4	471.6	1.408	40.84	1.162	16.23
		min	3219.0	421.0	4200.0	483.0	1.383	38.29	1.147	14.73	4550.0	462.0	1.567	56.65	1.170	16.96	4364.0	541.0	1.499	49.86	1.313	31.34
		max	2826.0	384.0	3438.0	397.0	1.169	16.90	0.978	-2.22	4295.0	397.0	1.378	37.81	0.978	-2.22	4078.0	431.0	1.335	33.52	1.046	4.61
	5	avg	3232.6	535.5	3939.7	542.5	1.219	21.91	1.014	1.40	4564.8	646.8	1.412	41.24	1.208	20.81	4469.7	592.7	1.383	38.30	1.108	10.81
		min	3297.0	555.0	4167.0	591.0	1.289	28.93	1.092	9.24	4629.0	693.0	1.451	45.11	1.277	27.73	4732.0	657.0	1.461	46.09	1.214	21.44
		max	3097.0	490.0	3774.0	489.0	1.155	15.48	0.881	-11.89	4408.0	586.0	1.357	35.69	1.065	6.49	4331.0	525.0	1.336	33.63	0.946	-5.41
30x20x[21.25]	2	avg	3218.3	882.2	3801.7	901.7	1.183	18.35	1.026	2.56	4276.4	844.5	1.331	33.05	0.959	-4.15	4214.2	949.0	1.312	31.16	1.079	7.91
		min	3512.0	940.0	3961.0	1025.0	1.279	27.90	1.236	23.64	4479.0	931.0	1.397	39.67	1.022	2.20	4377.0	1057.0	1.387	38.71	1.203	20.25
		max	3007.0	821.0	3655.0	828.0	1.105	10.54	0.883	-11.73	4146.0	815.0	1.271	27.07	0.905	-9.47	4080.0	884.0	1.228	22.78	0.942	-5.76
	3	avg	3337.9	986.9	4128.6	1081.3	1.240	23.98	1.096	9.57	4533.4	1012.6	1.361	36.11	1.027	2.69	4385.3	1050.3	1.316	31.60	1.064	6.43
		min	3549.0	1028.0	4518.0	1144.0	1.352	35.15	1.161	16.14	4600.0	1048.0	1.457	45.66	1.105	10.55	4560.0	1174.0	1.408	40.77	1.169	16.93
		max	3075.0	948.0	3773.0	986.0	1.072	7.19	1.040	4.01	4441.0	987.0	1.284	28.37	0.993	-0.68	4180.0	963.0	1.238	23.77	0.963	-3.70
	5	avg	3902.0	1280.4	4381.8	1311.1	1.124	12.44	1.025	2.53	4726.5	1465.7	1.214	21.44	1.145	14.48	4608.2	1286.7	1.184	18.38	1.007	0.67
		min	4270.0	1349.0	4776.0	1426.0	1.205	20.46	1.130	13.03	4943.0	1551.0	1.348	34.78	1.206	20.60	4780.0	1492.0	1.272	27.21	1.223	22.30
		max	3539.0	1220.0	3898.0	1166.0	0.999	-0.10	0.868	-13.18	4548.0	1351.0	1.146	14.64	1.069	6.90	4401.0	1182.0	1.119	11.94	0.940	-5.96

Table 6.42 (Cont'd)

			MRHP_SR		MRHP_BMCH		BMCH - SR comparison				MRHP_FJOB		FJOB - SR comparison				MRHP_BJOB		BJOB - SR comparison			
							C_{max}		time				C_{max}		time				C_{max}		time	
	w		C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)
50x10x[10.15]	2	avg	4085.6	320.4	5946.3	341.6	1.456	45.60	1.081	8.09	6695.7	288.8	1.640	63.95	0.913	-8.65	6373.6	395.4	1.561	56.07	1.241	24.09
		min	4289.0	422.0	6235.0	382.0	1.572	57.17	1.295	29.49	6948.0	303.0	1.736	73.61	1.032	3.18	6568.0	483.0	1.649	64.86	1.407	40.68
		max	3967.0	283.0	5446.0	315.0	1.333	33.32	0.822	-17.77	6302.0	282.0	1.543	54.27	0.682	-31.75	5936.0	318.0	1.453	45.31	1.099	9.86
	3	avg	4221.1	337.1	6078.8	381.6	1.440	44.03	1.134	13.41	6999.7	333.3	1.660	65.97	0.990	-1.05	6671.3	415.5	1.581	58.09	1.230	22.96
		min	4376.0	352.0	6522.0	503.0	1.527	52.67	1.506	50.60	7185.0	343.0	1.781	78.14	1.061	6.05	6994.0	682.0	1.645	64.53	1.938	93.75
		max	3929.0	314.0	5509.0	343.0	1.317	31.70	0.980	-2.00	6767.0	310.0	1.547	54.68	0.949	-5.13	6236.0	345.0	1.518	51.79	1.006	0.57
	5	avg	4697.5	411.2	6069.0	459.6	1.324	32.44	1.118	11.85	7425.0	524.6	1.624	62.42	1.276	27.62	7237.1	468.4	1.582	58.18	1.140	13.98
		min	6385.0	438.0	6398.0	495.0	1.498	49.80	1.172	17.24	7889.0	759.0	1.814	81.44	1.847	84.67	7491.0	511.0	1.810	81.02	1.178	17.77
		max	3998.0	377.0	5564.0	442.0	0.935	-6.55	1.067	6.73	7115.0	452.0	1.141	14.14	1.175	17.52	6749.0	444.0	1.133	13.34	1.068	6.78
50x20x[21.25]	2	avg	4561.0	2156.3	5867.2	2350.2	1.287	28.70	1.090	9.04	6903.7	2299.8	1.514	51.45	1.068	6.76	6714.4	2422.7	1.473	47.26	1.125	12.50
		min	4943.0	2281.0	6075.0	2470.0	1.332	33.22	1.146	14.56	7264.0	2487.0	1.593	59.26	1.138	13.77	7044.0	2869.0	1.514	51.44	1.365	36.49
		max	4404.0	2058.0	5528.0	2221.0	1.229	22.90	1.026	2.59	6689.0	2121.0	1.432	43.23	0.961	-3.94	6408.0	2280.0	1.405	40.50	1.038	3.80
	3	avg	4648.6	2503.7	6283.9	2751.2	1.352	35.20	1.102	10.22	7267.7	2637.4	1.564	56.37	1.056	5.61	7031.1	3086.6	1.511	51.07	1.236	23.57
		min	4748.0	2761.0	6640.0	2894.0	1.429	42.86	1.242	24.25	7445.0	2877.0	1.631	63.13	1.208	20.77	7306.0	3271.0	1.561	56.11	1.358	35.76
		max	4540.0	2326.0	5840.0	2392.0	1.254	25.40	0.950	-4.97	6962.0	2455.0	1.477	47.69	0.926	-7.43	6745.0	2794.0	1.462	46.21	1.120	11.96
	5	avg	5135.1	3263.1	6385.2	3556.3	1.244	24.40	1.092	9.21	7563.3	3954.4	1.474	47.37	1.214	21.41	7499.2	3580.4	1.461	46.10	1.099	9.88
		min	5361.0	3493.0	6705.0	3873.0	1.336	33.59	1.206	20.58	8005.0	4131.0	1.595	59.49	1.321	32.15	7798.0	3702.0	1.530	52.98	1.180	18.02
		max	4940.0	3108.0	6071.0	3380.0	1.182	18.23	0.968	-3.24	7174.0	3778.0	1.397	39.71	1.091	9.13	7037.0	3386.0	1.370	37.04	1.025	2.49
100x10x[10.15]	2	avg	7691.4	1089.4	11561.3	1249.0	1.503	50.34	1.147	14.67	13511.3	1167.1	1.757	75.74	1.072	7.16	12360.4	1476.9	1.607	60.70	1.356	35.59
		min	7997.0	1127.0	12014.0	1308.0	1.573	57.34	1.180	18.01	13734.0	1390.0	1.840	83.95	1.276	27.64	12832.0	1539.0	1.668	66.84	1.413	41.32
		max	7466.0	1033.0	11122.0	1209.0	1.464	46.39	1.097	9.75	13285.0	1066.0	1.670	67.03	0.979	-2.11	11551.0	1418.0	1.544	54.40	1.307	30.67
	3	avg	7849.8	1352.2	12342.1	1420.6	1.572	57.24	1.059	5.90	13778.1	1310.6	1.756	75.55	0.980	-2.04	13212.2	1606.5	1.683	68.34	1.202	20.16
		min	7987.0	1483.0	12701.0	1557.0	1.616	61.61	1.253	25.26	14203.0	1376.0	1.818	81.82	1.111	11.12	13504.0	1809.0	1.743	74.35	1.462	46.24
		max	7578.0	1169.0	12026.0	1310.0	1.543	54.35	0.904	-9.64	13201.0	1243.0	1.680	67.97	0.838	-16.18	12215.0	1449.0	1.554	55.43	0.977	-2.29
	5	avg	7859.3	1483.1	12159.8	1778.7	1.548	54.77	1.200	19.97	14696.2	1931.1	1.870	87.05	1.303	30.26	14062.5	1837.5	1.790	78.98	1.239	23.89
		min	8163.0	1542.0	12589.0	1967.0	1.639	63.86	1.326	32.64	14944.0	2134.0	1.944	94.38	1.424	42.36	14360.0	2024.0	1.868	86.82	1.356	35.57
		max	7564.0	1408.0	11566.0	1569.0	1.472	47.17	1.058	5.80	14068.0	1817.0	1.790	79.00	1.178	17.83	13303.0	1621.0	1.693	69.27	1.093	9.31

Table 6.43 Summary of the results

			MRHP_SR		MRHP_BMCH		BMCH - SR comparison				MRHP_FJOB		FJOB - SR comparison				MRHP_BJOB		BJOB - SR comparison			
							C_{max}		time				C_{max}		time				C_{max}		time	
	w		C_{max}	time	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)	C_{max}	time	ratio	RE(%)	ratio	RE(%)
10x10x[11.15]	**2**	**avg**	1248.7	36.4	1389.7	36.1	1.119	11.86	1.000	-0.01	1480.2	34.4	1.191	19.15	0.952	-4.82	1453.1	35.3	1.171	17.11	0.977	-2.34
	3	**avg**	1343.8	37.6	1405.1	39.6	1.054	5.43	1.055	5.48	1550.7	37.8	1.163	16.33	1.007	0.67	1496.2	37.8	1.122	12.17	1.009	0.85
10x15x[16.20]	**2**	**avg**	1432.7	72.0	1521.0	77.1	1.062	6.18	1.077	7.69	1581.5	68.0	1.105	10.53	0.948	-5.19	1590.0	69.0	1.111	11.15	0.963	-3.75
	3	**avg**	1650.7	76.9	1615.6	83.7	0.981	-1.94	1.089	8.86	1685.1	77.7	1.022	2.24	1.012	1.16	1680.7	78.1	1.020	2.00	1.017	1.70
10x20x[21.25]	**2**	**avg**	1663.1	141.3	1639.7	155.0	0.986	-1.38	1.098	9.84	1728.8	151.7	1.040	3.99	1.075	7.46	1692.2	159.2	1.017	1.74	1.127	12.70
	3	**avg**	1937.3	149.7	1839.6	167.8	0.950	-5.01	1.123	12.29	1740.4	177.0	0.899	-10.10	1.186	18.62	1739.3	160.4	0.898	-10.18	1.075	7.51
20x10x[11.15]	**2**	**avg**	2045.2	67.1	2556.2	77.3	1.251	25.08	1.153	15.35	2763.8	67.5	1.352	35.19	1.007	0.66	2702.7	75.4	1.324	32.36	1.124	12.41
	3	**avg**	2031.0	77.8	2534.2	81.6	1.250	25.02	1.051	5.13	2858.1	78.3	1.410	40.95	1.008	0.83	2793.6	81.2	1.377	37.66	1.044	4.38
	5	**avg**	2242.4	99.0	2596.9	97.4	1.159	15.95	0.986	-1.41	3021.8	101.2	1.349	34.87	1.024	2.38	3037.1	99.2	1.356	35.56	1.003	0.28
20x15x[16.20]	**2**	**avg**	2173.8	177.7	2598.2	193.1	1.197	19.67	1.089	8.90	2872.3	184.0	1.323	32.31	1.038	3.79	2837.3	199.1	1.307	30.71	1.122	12.24
	3	**avg**	2264.4	211.2	2706.1	220.0	1.197	19.66	1.044	4.40	3091.9	210.2	1.367	36.70	1.000	0.01	2905.5	226.0	1.285	28.46	1.075	7.55
	5	**avg**	2672.5	278.2	2894.2	256.2	1.084	8.44	0.928	-7.22	3176.3	301.4	1.190	19.04	1.091	9.13	3121.8	266.3	1.170	16.96	0.965	-3.53
20x20x[21.25]	**2**	**avg**	2429.5	409.2	2672.1	424.7	1.101	10.07	1.039	3.92	2963.0	407.8	1.220	22.00	0.997	-0.25	2939.8	439.6	1.210	21.04	1.075	7.50
	3	**avg**	2643.1	483.1	2895.4	508.6	1.097	9.67	1.052	5.23	3112.1	506.6	1.179	17.89	1.049	4.88	3099.3	531.0	1.175	17.47	1.100	9.96
	5	**avg**	3093.7	626.5	3230.3	619.9	1.045	4.55	0.991	-0.93	3247.7	687.9	1.051	5.14	1.098	9.83	3143.7	622.3	1.018	1.79	0.994	-0.59
30x10x[11.15]	**2**	**avg**	2721.4	120.7	3689.1	134.5	1.357	35.65	1.114	11.44	4082.8	119.6	1.503	50.27	0.992	-0.84	3874.3	148.0	1.426	42.63	1.226	22.64
	3	**avg**	2783.7	139.9	3749.5	153.8	1.348	34.79	1.099	9.91	4222.8	155.7	1.518	51.84	1.115	11.46	4073.0	152.7	1.465	46.49	1.092	9.19
	5	**avg**	3012.8	181.0	3875.6	178.4	1.290	29.03	0.987	-1.26	4483.7	197.9	1.493	49.34	1.093	9.33	4402.2	183.7	1.467	46.65	1.016	1.61
30x15x[16.20]	**2**	**avg**	2930.5	353.0	3690.1	387.6	1.260	26.03	1.099	9.94	4064.1	355.4	1.388	38.80	1.009	0.87	4036.4	407.8	1.379	37.87	1.157	15.71
	3	**avg**	3010.3	406.0	3843.0	432.3	1.277	27.69	1.065	6.48	4427.0	442.1	1.472	47.24	1.090	8.97	4235.4	471.6	1.408	40.84	1.162	16.23
	5	**avg**	3232.6	535.5	3939.7	542.5	1.219	21.91	1.014	1.40	4564.8	646.8	1.412	41.24	1.208	20.81	4469.7	592.7	1.383	38.30	1.108	10.81
30x20x[21.25]	**2**	**avg**	3218.3	882.2	3801.7	901.7	1.183	18.35	1.026	2.56	4276.4	844.5	1.331	33.05	0.959	-4.15	4214.2	949.0	1.312	31.16	1.079	7.91
	3	**avg**	3337.9	986.9	4128.6	1081.3	1.240	23.98	1.096	9.57	4533.4	1012.6	1.361	36.11	1.027	2.69	4385.3	1050.3	1.316	31.60	1.064	6.43
	5	**avg**	3902.0	1280.4	4381.8	1311.1	1.124	12.44	1.025	2.53	4726.5	1465.7	1.214	21.44	1.145	14.48	4608.2	1286.7	1.184	18.38	1.007	0.67
50x10x[10.15]	**2**	**avg**	4085.6	320.4	5946.3	341.6	1.456	45.60	1.081	8.09	6695.7	288.8	1.640	63.95	0.913	-8.65	6373.6	395.4	1.561	56.07	1.241	24.09
	3	**avg**	4221.1	337.1	6078.8	381.6	1.440	44.03	1.134	13.41	6999.7	333.3	1.660	65.97	0.990	-1.05	6671.3	415.5	1.581	58.09	1.230	22.96
	5	**avg**	4697.5	411.2	6069.0	459.6	1.324	32.44	1.118	11.85	7425.0	524.6	1.624	62.42	1.276	27.62	7237.1	468.4	1.582	58.18	1.140	13.98
50x20x[21.25]	**2**	**avg**	4561.0	2156.3	5867.2	2350.2	1.287	28.70	1.090	9.04	6903.7	2299.8	1.514	51.45	1.068	6.76	6714.4	2422.7	1.473	47.26	1.125	12.50
	3	**avg**	4648.6	2503.7	6283.9	2751.2	1.352	35.20	1.102	10.22	7267.7	2637.4	1.564	56.37	1.056	5.61	7031.1	3086.6	1.511	51.07	1.236	23.57
	5	**avg**	5135.1	3263.1	6385.2	3556.3	1.244	24.40	1.092	9.21	7563.3	3954.4	1.474	47.37	1.214	21.41	7499.2	3580.4	1.461	46.10	1.099	9.88
100x10x[10.15]	**2**	**avg**	7691.4	1089.4	11561.3	1249.0	1.503	50.34	1.147	14.67	13511.3	1167.1	1.757	75.74	1.072	7.16	12360.4	1476.9	1.607	60.70	1.356	35.59
	3	**avg**	7849.8	1352.2	12342.1	1420.6	1.572	57.24	1.059	5.90	13778.1	1310.6	1.756	75.55	0.980	-2.04	13212.2	1606.5	1.683	68.34	1.202	20.16
	5	**avg**	7859.3	1483.1	12159.8	1778.7	1.548	54.77	1.200	19.97	14696.2	1931.1	1.870	87.05	1.303	30.26	14062.5	1837.5	1.790	78.98	1.239	23.89
average			**3326.4**	**628.7**	**4299.6**	**680.3**	**1.229**	**22.90**	**1.070**	**7.04**	**4881.7**	**690.3**	**1.376**	**37.62**	**1.061**	**6.06**	**4718.0**	**715.5**	**1.338**	**33.78**	**1.105**	**10.45**

Table 6.43 shows that the MRHP_SR outperforms the MRHP_FJOB, the MRHP_BMCH and the MRHP_BJOB in average based on solution quality. The MRHP_BMCH outperforms both the MRHP_FJOB and the MRHP_BJOB. Furthermore, the MRHP_BJOB outperforms the MRHP_FJOB.

Further analysis of Table 6.43 shows that, the MRHP_SR outperforms the MRHP_BMCH, the MRHP_FJOB, and the MRHP_BJOB based on computational time in average. Furthermore, the MRHP_FJOB outperforms both the MRHP_BMCH and the MRHP_BJOB, and the MRHP_BMCH outperforms the MRHP_BJOB.

It is interesting to note that the MRHP_BMCH, the MRHP_FJOB and the MRHP_BJOB outperform the MRHP_SR in some instances, especially for the three time windows for 10 jobs instances. In some of the instances especially for small size problems, the four rules retrieve similar solutions, whereas the MRHP_SR finds a better solution overall for 20, 30, 50, and 100 job instances.

The first expectation is that the quality of the solution decreases when the number of time windows involved in the decomposition increases. This was confirmed in most tests. As it is seen in Table 6.43, for all four rules the solution quality of two window is better than three window for all job-machine combinations except a few instances. Further analysis of Table 6.43 reveals that the solution quality of three time windows is better than five time windows.

Even if the first expectation is that C_{max} increases when the number of time windows is greater, in some cases, the solution obtained with three time windows is slightly better than the one obtained with two time windows. It is shown in Table 6.44 for 10x10x[11,15] for the MHRP_SR for test problems 3 and 7.

Table 6.44 The results of the MHRP_SR for the 10x10x[11,15] instances

	MRHP_SR			
	window 2		**window 3**	
prb.	**C_{max}**	**time**	**C_{max}**	**time**
1	1198	35	1243	43
2	1184	37	1311	37
3	**1336**	37	**1313**	37
4	1232	37	1335	40
5	1239	45	1380	35
6	1313	35	1446	37
7	**1135**	34	**1127**	34
8	1559	34	1660	38
9	1133	34	1417	38
10	1158	36	1206	37

Similarly, experiments show that the MRHP_FJOB, the MRHP_BMCH and the MRHP_BJOB might give, in few cases, a better solution than the one found by the MRHP_SR especially for small size problems as it is seen in Table 6.45 for 10x15x[16,20] instances. It is probably because of the machine sequences of the jobs.

Table 6.45 The results of three time windows for 10x15x[16,20] instances

	MRHP_SR		**MRHP_BMCH**		**MRHP_FJOB**		**MRHP_BJOB**	
prb	**C_{max}**	**time**	**C_{max}**	**time**	**C_{max}**	**time**	**C_{max}**	**time**
1	1728	76	1741	81	1624	82	1496	71
2	1686	78	1568	84	1609	80	1694	76
3	1708	77	1490	83	1729	82	1742	91
4	1650	77	1618	78	1729	74	1672	82
5	1462	79	1530	88	1580	78	1589	73
6	1712	75	1679	86	1869	77	1813	87
7	1692	83	1497	90	1711	75	1743	74
8	1685	74	1832	81	1798	74	1798	74
9	1718	76	1673	86	1656	76	1706	80
10	1466	74	1528	80	1546	79	1554	73
avg.	**1650.7**	76.9	**1615.6**	83.7	**1685.1**	77.7	**1680.7**	78.1

Table 6.46 shows the average CPU times to solve the problems via the MRHP_SR with two and three time windows. It is expected that the CPU time decreases when the number of time windows increases. But, it is not, probably because of the sequence based algorithm of the MSBH_TML which is used in the MRHP, as it is seen in large size problems in Section 6.3. Furthermore, it is worth to mention that, the average time for 100 jobs – 10 machines problems are about 1000 seconds. This

value shows that large size problems in real world can be solve in reasonably short times with the MRHP_SR.

Table 6.46 The solution time results

	average solution times	
	window 2	**window 3**
10x10x[11,15]	36.4	37.6
10x15x[16,20]	72.0	76.9
10x20x[21,25]	141.3	149.7
20x10x[11,15]	67.1	77.8
20x15x[16,20]	177.7	211.2
20x20x[21,25]	409.2	483.1
30x10x[11,15]	120.7	139.9
30x15x[16,20]	353.0	406.0
30x20x[21,25]	882.2	986.9
50x10x[11,15]	320.4	337.1
50x20x[21,25]	2156.3	2503.7
100x10x[11,15]	1089.4	1352.2

Finally, we compared the results to evaluate the differences using the paired t-test at the 1% level of significance. Table 6.47 presents summary of all the results. In Table 6.47, the results whose difference is not statistically significant are highlighted in bold type. The difference of the results of the MRHP_BMCH, the MRHP_FJOB and the MRHP_BJOB with the MRHP_SR are statistically significant based on solution quality. The time difference is not statistically significant for the MRHP_FJOB for two and three time windows, and for the MRHP_BMCH and the MRHP_BJOB for five time windows.

Table 6.47 Summary of the results

	window 2		**window 3**		**window 5**	
	C_{max}	**time**	**C_{max}**	**time**	**C_{max}**	**time**
MRHP_BMCH – MRHP_SR	1.230	1.085	1.230	1.081	1.227	**1.038**
MRHP_FJOB – MRHP_SR	1.364	**1.002**	1.364	**1.043**	1.409	1.161
MRHP_BJOB –MRHP_ SR	1.325	1.131	1.320	1.109	1.379	**1.063**

6.6 Conclusion

In this chapter, we have evaluated the performance of several variations of the MSBH through extensive computational experiments on an extensive set of randomly generated test problems of reentrant job shops for makespan minimization objective.

The computational experiments showed that our modified methods were effective for the RJSSP. We prove that our modified L_{max} algorithm is effective for large instances, whereas the computational burden of the original SBH using exact subproblem solutions and full reoptimization increase rapidly with the number of operations. The original SBH is probably impractical for problems with more than about 500 operations. As a matter of fact, the size of the RJSSP instances becomes larger and larger in practice, the proposed methods seems able to meet the practical demands for both a quite high quality of solutions and a moderate computing time. However, for the larger instances the relation between time and quality is quite good.

Our computational experience shows that the performance of the MSBH (computation time and solution quality) is affected by the machine routing of the jobs especially for large instances. The actual CPU time savings depend on the routing structures. It is expected that for the same number of jobs and machines, the job with less operations is solved in less time than the job with more operations. But, in some instances, it is not like that because of the routing.

Subproblem prioritization procedures play an important role in the MSBH. Calculation of measures to determine the importance of a machine is an important direction. If the MSBH can recognize the importance of the machines and always schedule the critical machines first, the performance could be further improved.

Subproblem solution procedures have a significant effect on both solution quality and computation time. Hence a good schedule on the bottleneck machine is the key to the performance of the overall schedule. A poor subproblem solution may lead to

choosing the wrong machine as bottleneck early in the solution process, which will lead to a poor overall solution later on. Additionally, the solution obtained for a given subproblem at a given iteration of the procedure defines the release times, due dates and delayed precedence constraints of the subproblems solved at subsequent iterations. Poor subproblem solutions will thus lead to the wrong subproblem being defined for later stages of the solution process. Finally, a poor subproblem solution procedure will limit the usefulness of the resequencing step.

From our experiments, we can conclude that the MSBH works better when well-defined bottleneck machines are present in the shop, and that problems become easier as the capacity difference between bottleneck and non-bottleneck machines increases. When the bottleneck machines are scheduled well, the schedule of other non-bottleneck machines will easily fall into place. Hence, the MSBH has less need to reschedule, reducing both the number of subproblems solved and the time.

The results of job-machine sets shows that the solution time increases with the number of jobs, and with the number of machines. But, the increase in the number of machines increases solution times more than the increase in the number of jobs with a constant number of machine. Generally, the more subproblem solved, the longer CPU times required.

The result of the experiments shows that the performance of MSBH depends on the amount of resequencing. Resequencing step has a significant effect on both solution quality and time. Generally, the more computational effort the better the performance. The trade-off between solution time and quality is an important point.

The MSBH methods consistently outperform the MRHP on solution quality. For large problems, especially more than 50 jobs, the MSBH obtained the best solutions, but it spent too much CPU time than the MRHP.

In the next chapter, the application problem of the textile factory is described and solved by the modified methods.

CHAPTER SEVEN
APPLICATION PROBLEM IN A TEXTILE FACTORY

In the previous chapters, the reentrant job shop scheduling problem (RJSSP) and the proposed solution approaches for this problem were described in detail. In this chapter, the RJSSP of a textile factory is described and solved by the proposed methods.

The factory produces a large variety of manufactured fabrics, and works under a make-to-order policy in a complex process ranging from yarn producing to yarn dyeing, weaving and dyeing-finishing. We schedule the dyeing-finishing facility of the factory which is a special version of the RJSSP, and the scheduling horizon is one month.

7.1 Company Profile

The company is a vertically integrated textile company manufacturing yarn-dyed and piece-dyed shirting and sportswear fabrics. The plant employs 1400 workers and produces approximately 15.000.000 meters of fabric annually in its yarn, yarn-dye, weaving and dyeing-finishing facilities.

All manufacturing is based on customers' orders, and the factory works full capacity 24 hours a day and 330~335 days in a year. Main products are cotton yarn, raw and finished fabric.

There is an integrated manufacturing system. They produce yarn from cotton, and fabric from yarn. The product specifications come with the customer orders, and the manufacturing process is determined according to these specifications. The factory has different types of machines. Durations for operations are determined based on speed and productivity of machinery. Transportation activities and work-in-process inventory in the factory are high. Labourship is qualified. Planning department

handles the scheduling of orders in the company, and gives approximate delivery dates to customers depending on the length and quality of orders.

The company intends to schedule the orders as soon as possible in order to satisfy the given delivery dates. Apart from the primary objective there are some secondary objectives to consider, such as minimizing work-in-process inventory, maximizing machine utilizations, and minimizing the intervals between consecutive operations of a job. So, in each facility, it is aimed to reduce the makespan of each order.

7.1.1 The Reentrant Job Shop Scheduling Problem In The Dyeing-Finishing Facility

We schedule the jobs in the dyeing-finishing facility which is a reentrant job shop. The other three facilities (yarn, yarn-dyeing, weaving) where each yarn or fabric goes through the same machines are flow shops. The structure basically corresponds to a RJSSP with some special characteristics. For example, a job is processed on chemistry washing, washing and mercerized machines successively, and then it goes back to the chemistry washing and washing machines to be processed once again. Whereas the processing steps for each fabric are different, all jobs are processed first on the rolling machine and last on the fabric control machine. The main objective is to minimize makespan, because it is required that the jobs be completed as soon as possible, and go through the next step of the production.

Raw fabrics which are produced using raw and painted yarns in the weaving facility are taken to the dyeing-finishing facility where they become finished products. The capacity of this facility is approximately 1.000.000 meters/month. There are 3 shifts in a day, and 7 working days in a week. The number of workers is the same throughout the day. Planning department takes weaved fabrics to shop every morning, and loads machines one by one, using First Come First Served (FCFS) rule. Thus, daily working plan of all machinery is determined. The aim is to process all the orders belonging to that day.

In this study, we prepare a one-month production schedule for the dyeing-finishing facility. Scheduling is done on a daily basis for four weeks. Every morning, the jobs coming to the shop are scheduled to the pertinent machines. Due to the fact that the number of the jobs that come into the shop in a day is numerous, not all the jobs can be finished within the same day. Therefore, the unfinished jobs are procrastinated to the next day. In the next day, the jobs lagging from the day before are prioritized and finished first. The planned completion times of the jobs that lagged from the day before on the machines are regarded as the release dates of the waiting jobs to be scheduled on the current day. There are 20 different machines. For each machine daily available processing time is deemed as 1440 (24 hours * 60 minutes) minutes/day. Since the jobs whose makespan surpass this value are procrastinated to the next day, job piling is inclined to increase in the shop towards the end of the scheduling horizon.

Set-up times, machine breakdowns and transportation times among the machines are attributed to the processing times. For this time loss, the machine speeds are calculated via multiplying the original speed of the machine by the productivity (%) of the machine. The productivity values have been calculated using the statistical data of the past scheduling periods obtained from the planning department. Then, the processing time of a fabric on a machine is computed via the length of the fabric (meter) divided by the machine speed (meter/minute).

Machines can only process one job at a time, and preemptions are not allowed. When scheduling the new orders in a day, the existing production plan is not modified and so these orders must be scheduled in a way that respects the previous machine workload. Jobs do not have an urgency level or weight. There are precedence constraints between operations, specifying the order of production. The first operation of all the jobs is the rolling operation which is performed on machine 1. The last operation of all the jobs is the fabric control operation which is performed on machine 20. No job uses all the machines, and the number of operations per job ranges between 8-17. The operations of a job can not interrupted or overlapped.

Daily workload is given in the Table 7.1 for 28 days. For a topical example, the measurements of the length of the fabrics in meters and machine sequence of the jobs for a day (day 13) are given in Appendix C in detail. Table 7.2 presents the machines.

Table 7.1 The daily workload of the facility

day	number of jobs	number of operations	number of machines	total length of fabrics (meter)	total processing times (minute)
1	54	544	16	53450	9354
2	72	787	18	52900	9256
3	48	566	18	32600	6207
4	41	458	19	50150	9434
5	27	281	17	36900	6155
6	10	104	12	29900	5059
7	38	382	13	50400	8543
8	55	615	18	55000	9859
9	66	689	17	52300	8887
10	39	410	13	31050	5230
11	38	414	19	35350	6232
12	44	464	19	56650	9904
13	20	239	19	26700	5001
14	44	443	16	58700	10062
15	42	435	19	35150	6059
16	26	264	15	35500	6012
17	12	147	18	30000	5825
18	10	112	13	28250	5010
19	15	164	16	38050	6509
20	22	230	13	40950	6991
21	41	413	17	46500	7419
22	43	457	14	48600	8431
23	33	345	11	58650	9931
24	34	356	12	50050	9087
25	61	628	12	56700	7162
26	33	336	12	42900	7070
27	20	228	16	39350	7405
28	12	119	12	35750	5874
total	1000	10630	434	1208450	207968

Table 7.2 Machines

machine	process
1	Rolling
2	Burning
3	Mercerized
4	Chemistry washing
5	Washing
6	Bleaching
7	Hot flue drying
8	Drying
9	Raising
10	Peach touch
11	Prescribtion control
12	Painting
13	Jet Painting
14	Color control
15	Optical color kontrol
16	Softening
17	Finishin and drying
18	Polishing
19	Unshrinkable
20	Fabric control

7.2 Solution Of The Application Problem

As described in Chapter 4 and 5, the modified shifting bottleneck heuristic (MSBH) and the modified rolling horizon procedure (MRHP) are proposed to produce good quality solutions in short computing times. Various randomly generated problems were tested using all the proposed methods in Chapter 6. We prove that the MSBH and the MSBH with Total Machine Load (MSBH_TML) give better solutions in less computational times than the MSBH with Total Infeasibility (MSBH_TOTINF). Furthermore, we show that, for this special RJSSP, the MRHP with special rule (MRHP_SR) outperforms the other rules for assignment of operations to time windows. So, in order to solve this problem, we use the MSBH and the MSBH_TML, and the MRHP_SR. In the MHRP_SR, the number of time windows has been taken as two. We prove in Section 6.4 that for less than 50 jobs the MRHP with three or more time windows does not give less solution time than that of with two time windows, due to the fact that the MSBH has a sequencing based algorithm. Furthermore, the methods have been applied with resequencing.

The number of resequencing cycles is fixed to one, because we see in Section 6.3 that more than one cycle does not give any significant improvement on solution quality, but increases computational times.

The scheduling is done on a daily basis for 28 days. Thus, the modified heuristics are tested on the orders with up to 72 jobs, 787 operations and 19 machines in a day. Table 7.3 gives the solution results. Solution quality and time for each instance are expressed as the ratio of C_{max} and CPU time obtained by one of the proposed methods to that of the other one. The relative error RE(%) was calculated for each day, as a percentage by which the solution obtained is above the value obtained by the other method.

From Table 7.3, for the MSBH_TML, we see that the average solution gap is 1.94% compared to the MSBH, and the procedure performs quite well in most of the days. Analysis of the table reveals that, the MSBH_TML solves 27 of 28 days within 5% solution gap of the MSBH. Furthermore, with the MSBH_TML, the solutions obtained for seventeenth and nineteenth days are less than the C_{max} values obtained by the MSBH.

The average solution gap of the MRHP_SR is within 0.44% of the MSBH, and -1.47% of the MSBH_TML. It is interesting to see that the MRHP_SR outperforms the MSBH_TML based on solution quality. It is probably because of the special characteristic of the shop. Furthermore, in the MRHP_SR, we arrange the jobs in increasing order of lengths. In 12 of 28 days, the MRHP_SR produces slightly better results than the MSBH.

Further analysis of Table 7.3 reveals that the MSBH_TML and the MRHP_SR solve the problems in less computational times on average than the MSBH. The MSBH_TML outperforms the other two methods on time basis. Average time gap is within -9.97% of the MSBH for the MSBH_TML. Furthermore, the MSBH_SR is outperformed by the MRHP_TML on time basis. Average time gap is within 10.89% of the MSBH_TML for the MSBH_SR.

Table 7.3 The solution results of the methods for the application problem

day	MSBH		MSBH_TML		MRHP_SR		MSBH_TML - MSBH				MHRP_SR - MSBH				MHRP_SR - MSBH_TML			
							Cmax		time		Cmax		time		Cmax		time	
	Cmax	time	Cmax	time	Cmax	time	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)	ratio	RE(%)
1	1798	491	1879	437	1823	426	1.045	4.51	0.890	-11.00	1.014	1.39	0.868	-13.24	0.970	-2.98	0.975	-2.52
2	2121	1013	2203	817	2147	825	1.039	3.87	0.807	-19.35	1.012	1.23	0.814	-18.56	0.975	-2.54	1.010	0.98
3	1768	544	1849	465	1793	527	1.046	4.58	0.855	-14.52	1.014	1.41	0.969	-3.13	0.970	-3.03	1.133	13.33
4	2000	376	2081	370	2025	356	1.041	4.05	0.984	-1.60	1.013	1.25	0.947	-5.32	0.973	-2.69	0.962	-3.78
5	1790	133	1871	117	1815	118	1.045	4.53	0.880	-12.03	1.014	1.40	0.887	-11.28	0.970	-2.99	1.009	0.85
6	1425	26	1428	21	1375	33	1.002	0.21	0.808	-19.23	0.965	-3.51	1.269	26.92	0.963	-3.71	1.571	57.14
7	1751	164	1784	135	1763	160	1.019	1.88	0.823	-17.68	1.007	0.69	0.976	-2.44	0.988	-1.18	1.185	18.52
8	2144	618	2178	587	2157	587	1.016	1.59	0.950	-5.02	1.006	0.61	0.950	-5.02	0.990	-0.96	1.000	0.00
9	2447	765	2481	677	2460	726	1.014	1.39	0.885	-11.50	1.005	0.53	0.949	-5.10	0.992	-0.85	1.072	7.24
10	2042	185	2076	150	2055	197	1.017	1.67	0.811	-18.92	1.006	0.64	1.065	6.49	0.990	-1.01	1.313	31.33
11	1780	330	1814	318	1793	348	1.019	1.91	0.964	-3.64	1.007	0.73	1.055	5.45	0.988	-1.16	1.094	9.43
12	2229	410	2263	361	2242	318	1.015	1.53	0.880	-11.95	1.006	0.58	0.776	-22.44	0.991	-0.93	0.881	-11.91
13	1679	113	1713	113	1692	132	1.020	2.03	1.000	0.00	1.008	0.77	1.168	16.81	0.988	-1.23	1.168	16.81
14	2195	296	2229	260	2208	263	1.015	1.55	0.878	-12.16	1.006	0.59	0.889	-11.15	0.991	-0.94	1.012	1.15
15	1927	330	1961	317	1940	301	1.018	1.76	0.961	-3.94	1.007	0.67	0.912	-8.79	0.989	-1.07	0.950	-5.05
16	1670	99	1704	88	1683	107	1.020	2.04	0.889	-11.11	1.008	0.78	1.081	8.08	0.988	-1.23	1.216	21.59
17	1326	59	1308	52	1307	61	0.986	-1.36	0.881	-11.86	0.986	-1.43	1.034	3.39	0.999	-0.08	1.173	17.31
18	1307	29	1375	29	1384	32	1.052	5.20	1.000	0.00	1.059	5.89	1.103	10.34	1.007	0.65	1.103	10.34
19	1528	56	1509	54	1509	53	0.988	-1.24	0.964	-3.57	0.988	-1.24	0.946	-5.36	1.000	0.00	0.981	-1.85
20	1560	75	1590	70	1558	85	1.019	1.92	0.933	-6.67	0.999	-0.13	1.133	13.33	0.980	-2.01	1.214	21.43
21	1670	271	1700	259	1668	212	1.018	1.80	0.956	-4.43	0.999	-0.12	0.782	-21.77	0.981	-1.88	0.819	-18.15
22	1850	271	1880	243	1848	248	1.016	1.62	0.897	-10.33	0.999	-0.11	0.915	-8.49	0.983	-1.70	1.021	2.06
23	2365	114	2395	99	2363	120	1.012	1.25	0.868	-13.16	0.999	-0.10	1.053	5.26	0.987	-1.34	1.212	21.21
24	2593	134	2623	119	2591	146	1.011	1.15	0.888	-11.19	0.999	-0.08	1.090	8.96	0.988	-1.22	1.227	22.69
25	2583	372	2613	272	2581	304	1.012	1.15	0.731	-26.88	0.999	-0.08	0.817	-18.28	0.988	-1.22	1.118	11.76
26	2573	123	2603	116	2571	142	1.012	1.16	0.943	-5.69	0.999	-0.08	1.154	15.45	0.988	-1.23	1.224	22.41
27	2445	86	2475	76	2443	91	1.012	1.21	0.884	-11.63	0.999	-0.09	1.058	5.81	0.987	-1.29	1.197	19.74
28	2196	24	2226	24	2194	29	1.014	1.38	1.000	0.00	0.999	-0.07	1.208	20.83	0.986	-1.44	1.208	20.83
avg.	**1956**	**268**	**1993**	**237**	**1964**	**248**	**1.019**	**1.94**	**0.900**	**-9.97**	**1.004**	**0.44**	**0.995**	**-0.47**	**0.985**	**-1.47**	**1.109**	**10.89**

Analysis of Table 7.3 reveals that in 25 of 28 days C_{max} values which are obtained by the MSBH are higher than 1440 minutes, so the jobs are procrastinated to the next day. However, none of the jobs are procrastinated to the day after the next day. The maximum makespan is 2593 minutes for day 24. In day 28, the makespan value is 2196. It means that the last operation of the last job which is processed on machine 20 is finished at time 756 in day 29. The average makespan value is 1956 minutes. It shows that the jobs are completed in approximately one and a half day. The data which is obtained from the factory shows that the average completion time of the jobs is two-three days.

The procrastinated operations that lagged from the day before are finished first the next day. Their sequences are determined the day before, so there is no need to sequence them again the next day. The jobs that belong to a particular day are scheduled to the machines according to the completion times of the operations that are lag from the day before on those machines.

In Figure 7.1, the comparison of the MSBH_TML with the MSBH is given based on solution quality. We see that, most of the solution ratios vary between 1.010 and 1.020. Figure 7.2 shows the comparison of the MSBH_TML with the MSBH based on solution time. For all of the days of the scheduling horizon, the MSBH_TML find the solutions in less or equal times than the MSBH.

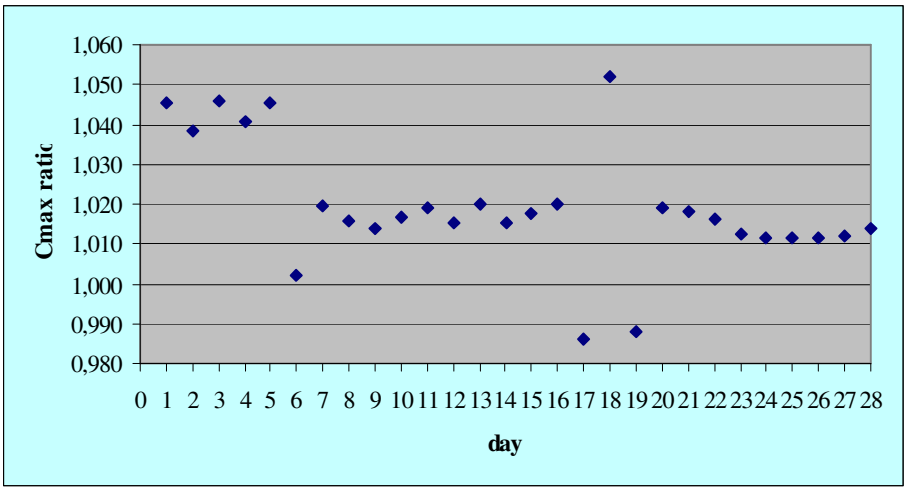


Figure 7.1 Comparison of MSBH_TML and MSBH based on quality

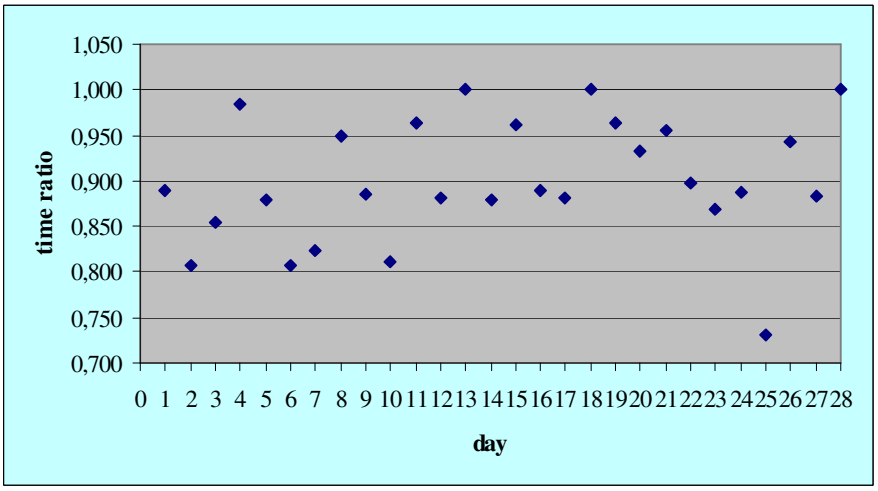


Figure 7.2. Comparison of MSBH_TML and MSBH based on time

In Figure 7.3, the comparison of the MRHP_SR with the MSBH is given based on solution quality. We see that, most of the solution ratios vary between 1.000 and 1.020 especially. Figure 7.4 shows the comparison of the MRHP_SR with the MSBH based on solution time. The time ratios vary between 0.800 and 1.200 especially.

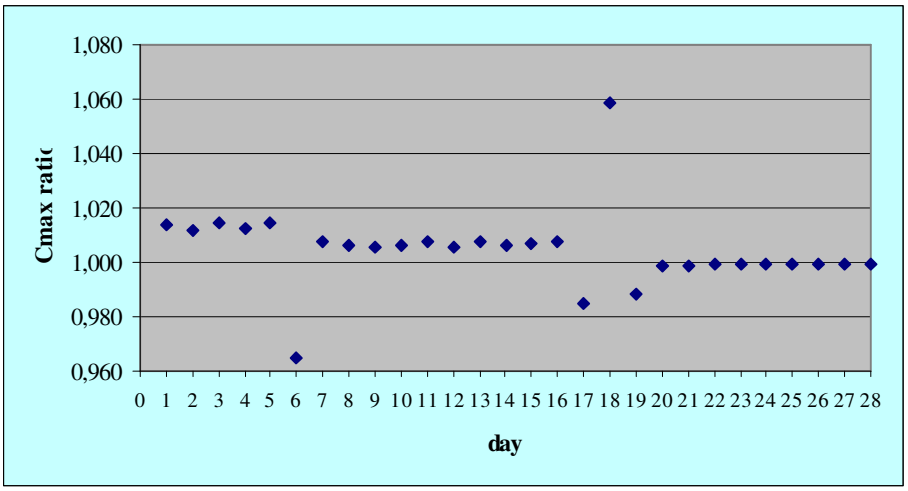


Figure 7.3 Comparison of MRHP_SR and MSBH based on quality

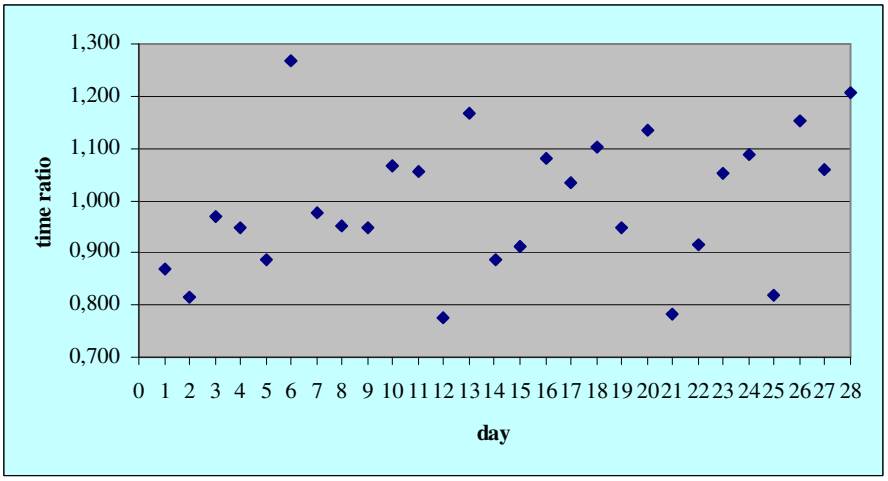


Figure 7.4 Comparison of MRHP_SR and MSBH based on time

In Figure 7.5, the comparison of the MRHP_SR with the MSBH_TML is given based on solution quality. In 26 of 28 days, the MSBH_SR outperforms the MSBH_TML. Figure 7.6 shows the comparison of the MRHP_SR and the MSBH_TML based on solution time. In most of the days, the MSBH_SR is outperformed by the MSBH_TML on time basis.

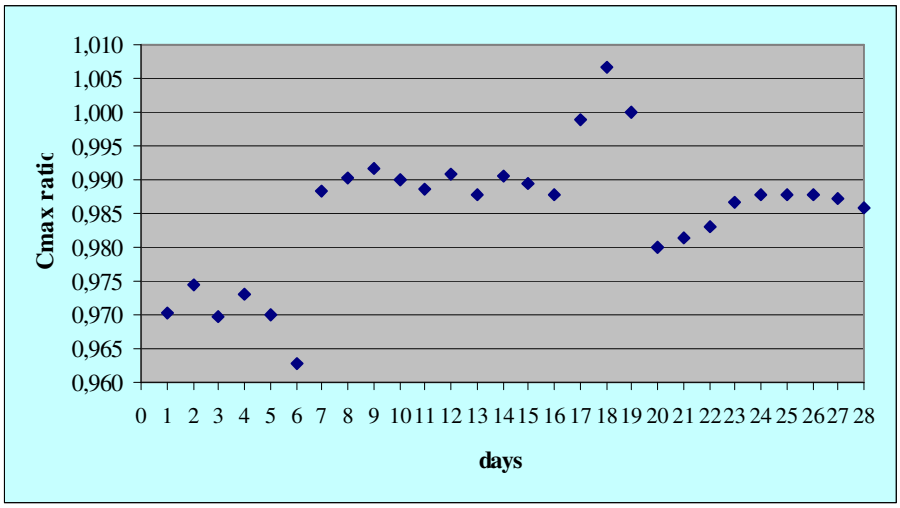


Figure 7.5 Comparison of MRHP_SR and MSBH_TML based on quality

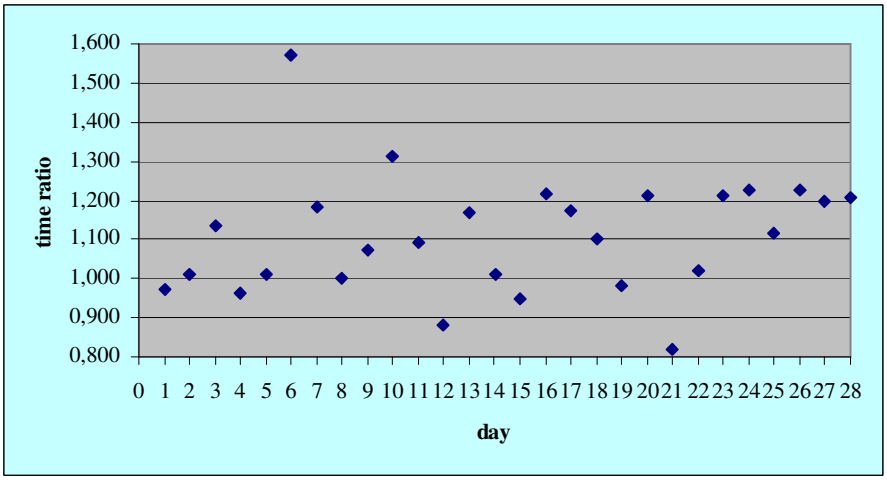


Figure 7.6 Comparison of MRHP_SR and MSBH_TML based on time

Tables 7.4, 7.5 and 7.6 show the times in minutes that were procrastinated to the next day for each machine for the MSBH, the MSBH_ TML, and the MRHP_SR successively. Especially machine 20 which is the last processing machine for all the jobs has too many procrastinated jobs.

Analysis of Table 7.4 reveals that, for machine 20, the average time is 525 minutes for 28 days. As it is seen in Table 7.4, the jobs could be finished on the day they belong on six of the machines (machines 1, 2, 9, 10, 11, 12). For example, on day 1, for machine 1, the total time of procrastinated operations is zero, so there are

not any procrastinated jobs to day 2. The average times which are given in the last line of Table 7.4 show the average procrastinated minutes for twenty machines. The machines which have zero procrastinated minutes in average are the machines that do not have any procrastinated operations. Table 7.5 shows that for machine 20, the average time is 561 minutes for the MSBH_TML, whereas it is 533 minutes for the MRHP_SR.

Table 7.7 shows the percentages of procrastinated operations and jobs for each day for the MSBH. On average 5.6% of operations, and 32.81% of jobs are procrastinated to the next day in the 28-day period. Figures 7.7 and 7.8 show the number of procrastinated operations and jobs.

Table 7.4 The total procrastinated minutes for each machine for the MSBH

	machine																			
day	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43	358
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	681
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	328
4	0	0	0	0	0	0	0	0	0	0	0	0	77	89	0	0	114	0	156	560
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	350
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	310
8	0	0	0	0	10	0	22	40	0	0	0	0	0	0	0	0	82	185	235	704
9	0	0	9	12	15	0	0	63	0	0	0	0	0	0	0	11	66	77	87	1007
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	602
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	340
12	0	0	0	0	0	0	65	0	0	0	0	0	0	0	80	0	110	176	226	789
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	239
14	0	0	115	62	87	82	244	7	0	0	0	0	0	0	0	0	274	341	391	755
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	487
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	230
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	230
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	80	410
23	0	0	334	341	348	327	0	395	0	0	0	0	0	0	0	0	431	521	571	925
24	0	0	392	356	383	385	416	410	0	0	0	0	0	0	0	0	443	541	584	1153
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1143
26	0	0	1	5	9	0	0	13	0	0	0	0	0	0	0	0	23	71	101	1133
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1005
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	756
avg.	**0.0**	**0.0**	**30.4**	**27.7**	**30.4**	**28.3**	**26.7**	**33.2**	**0.0**	**0.0**	**0.0**	**0.0**	**2.7**	**3.2**	**2.8**	**0.4**	**55.1**	**69.4**	**89.1**	**525.0**

Table 7.5 The total procrastinated minutes for each machine for the MSBH_TML

	machine																			
day	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	118	168	439
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	763
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	409
4	0	0	0	0	0	0	0	0	0	0	0	0	152	164	0	0	189	0	231	641
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	431
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	344
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	52	102	738
9	0	0	29	32	35	0	14	74	0	0	0	0	0	0	16	33	77	90	103	1041
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	636
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	374
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	91	141	823
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	273
14	0	0	0	0	0	0	93	0	0	0	0	0	0	0	0	0	123	190	240	789
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	521
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	264
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	69
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	149
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	259
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	439
23	0	0	514	545	581	507	0	611	0	0	0	0	0	0	0	0	641	708	758	954
24	0	0	282	249	276	275	306	303	0	0	0	0	0	0	0	0	337	435	480	1183
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1173
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	44	1163
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1035
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	786
avg.	**0**	**0**	**29**	**30**	**32**	**28**	**15**	**35**	**0**	**0**	**0**	**0**	**5**	**6**	**1**	**1**	**51**	**61**	**81**	**561**

Table 7.6 The total procrastinated minutes for each machine for the MRHP_SR

	machine																			
day	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	45	383
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	707
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	353
4	0	0	0	0	0	0	0	0	0	0	0	0	162	175	0	0	200	0	261	585
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	375
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	323
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58	108	717
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	52	99	1.02
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	615
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	353
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104	154	802
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	252
14	0	0	3	0	0	0	132	0	0	0	0	0	0	0	0	0	162	229	279	768
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	500
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	243
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	69
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	118
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	228
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	29	408
23	0	0	148	181	211	114	0	241	0	0	0	0	0	0	0	0	271	409	459	923
24	0	0	62	84	104	7	186	124	0	0	0	0	0	0	0	0	216	374	424	1.151
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.141
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.131
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	1.003
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	754
avg.	**0**	**0**	**8**	**9**	**11**	**4**	**11**	**13**	**0**	**0**	**0**	**0**	**6**	**6**	**0**	**0**	**30**	**44**	**67**	**533**

Table 7.7 The percentages of procrastinated operations and jobs for each day for the MSBH

	machine																				number of procrastinated operations	total daily operations	Percentage of procrastinated operations (%)	number of procrastinated job	total jobs	Percentage of procrastinated jobs (%)
day	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**						
1																			1	4	**5**	**544**	**0.92**	**4**	**54**	**7.41**
2																				27	**27**	**787**	**3.43**	**27**	**72**	**37.50**
3																				18	**18**	**566**	**3.18**	**18**	**48**	**37.50**
4													2	2			2		2	7	**15**	**458**	**3.28**	**7**	**41**	**17.07**
5																				9	**9**	**281**	**3.20**	**9**	**27**	**33.33**
6																					**0**	**104**	**0.00**	**0**	**10**	**0.00**
7																			1	4	**5**	**382**	**1.31**	**4**	**38**	**10.53**
8					1		1	1									2	3	3	14	**25**	**615**	**4.07**	**14**	**55**	**25.45**
9			1	2	3			3								1	5	7	10	42	**74**	**689**	**10.74**	**42**	**66**	**63.64**
10																				26	**26**	**410**	**6.34**	**26**	**39**	**66.67**
11																				9	**9**	**414**	**2.17**	**9**	**38**	**23.68**
12							1								1		2	3	3	9	**19**	**464**	**4.09**	**9**	**44**	**20.45**
13																				5	**5**	**239**	**2.09**	**5**	**20**	**25.00**
14			2	1	1	2	2	1									3	3	4	8	**27**	**443**	**6.09**	**8**	**44**	**18.18**
15																				19	**19**	**435**	**4.37**	**19**	**42**	**45.24**
16																				5	**5**	**264**	**1.89**	**5**	**26**	**19.23**
17																					**0**	**147**	**0.00**	**0**	**12**	**0.00**
18																					**0**	**112**	**0.00**	**0**	**10**	**0.00**
19																				1	**1**	**164**	**0.61**	**1**	**15**	**6.67**
20																				2	**2**	**230**	**0.87**	**2**	**22**	**9.09**
21																				3	**3**	**413**	**0.73**	**3**	**41**	**7.32**
22																		1	1	7	**9**	**457**	**1.97**	**7**	**43**	**16.28**
23			16	4	4	8		11									11	10	12	16	**92**	**345**	**26.67**	**16**	**33**	**48.48**
24			24	4	5	10	10	4									14	13	15	25	**124**	**356**	**34.83**	**25**	**34**	**73.53**
25																				55	**55**	**628**	**8.76**	**55**	**61**	**90.16**
26			1	1	1			2									3	3	4	23	**38**	**336**	**11.31**	**23**	**33**	**69.70**
27																				16	**16**	**228**	**7.02**	**16**	**20**	**80.00**
28																				8	**8**	**119**	**6.72**	**8**	**12**	**66.67**
avg.																					**22.71**	**379.64**	**5.60**	**12.93**	**35.71**	**32.81**

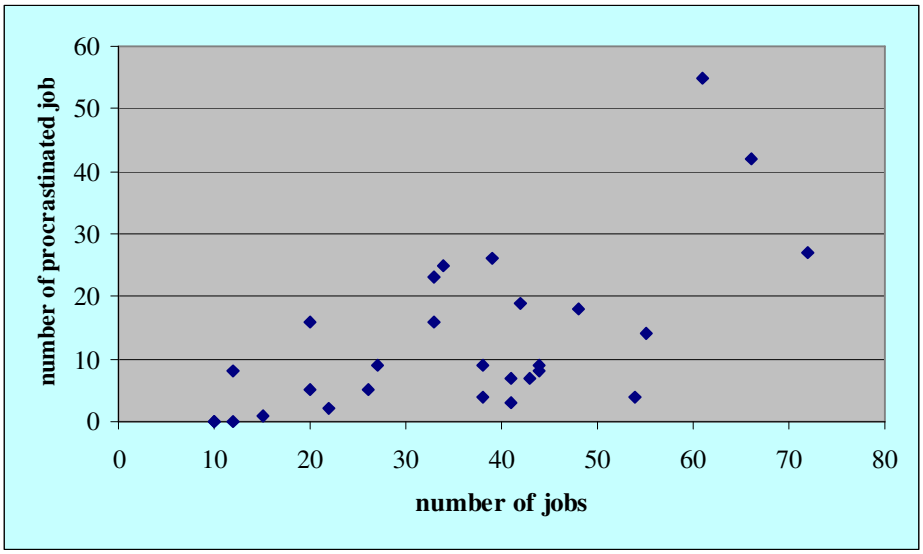


Figure 7.7 Procrastinated jobs

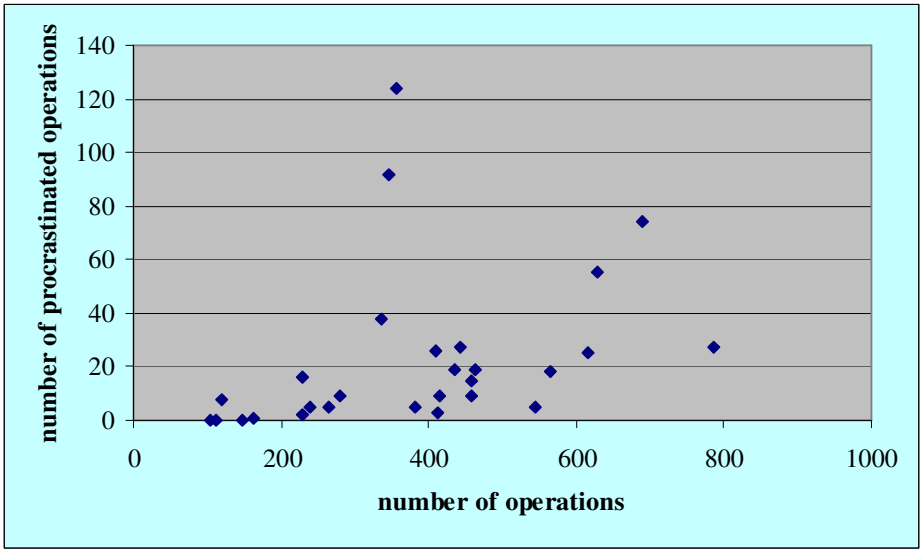


Figure 7.8 Procrastinated operations

Finally, we compared the results to evaluate the differences using the paired t-test at the 1% level of significance. In Table 7.7, the results whose difference is not statistically significant are highlighted in bold type. Analysis of Table 7.7 reveals that, on both solution time and quality, the difference between the MSBH and the MRHP_SR is not statistically significant.

Table 7.8 Average ratios for the methods

	C_{max}	time
MSBH_TML - MSBH	1.019	0.900
MRHP_SR - MSBH	**1.004**	**0.995**
MRHP_SR - MSBH_TML	0.985	1.109

The results show that the MSBH works well in this RJSSP. The MSBH and the MSBH_SR are quite similar, in terms of satisfying minimizing makespan. The MSBH_TML works slightly better than the other methods based on solution times. One of the objectives is to asses the quality of the solutions and the speed of methods. The tests show that the proposed methods are able to provide tight completion times for incoming customer orders involving more than 50 jobs in a few minutes.

Furthermore, the MSBH can evaluate the conditions better than the MSBH_TML, because it chooses the bottleneck in each iteration. Therefore, the MSBH_TML is effected from the procrastinated operations more than the MSBH.

We schedule the jobs that belong to one month, day by day. If there is a change in the jobs for a day, we can revise the schedule to the very day. For this rescheduling, we only need jobs and operations data and machines usages.

To see the schedule of a job in a day, Table 7.9 shows the data and Figure 7.9 shows the Gannt diagram of a job which belongs to day 13. For the same day, the schedule of 20 jobs on machine 3 is given in Figure 7.10, and the data is given in Table 7.10.

Table 7.9 The start and finish times of the operations for a job

job	operation	machine	processing time	start time	finish time
6	1	1	19	106	125
6	2	2	25	172	197
6	3	3	17	197	214
6	4	6	38	214	252
6	5	8	15	273	288
6	6	3	17	315	332
6	7	8	15	333	348
6	8	17	15	351	366
6	9	18	33	367	400
6	10	17	15	400	415
6	11	19	25	429	454
6	12	20	50	949	999

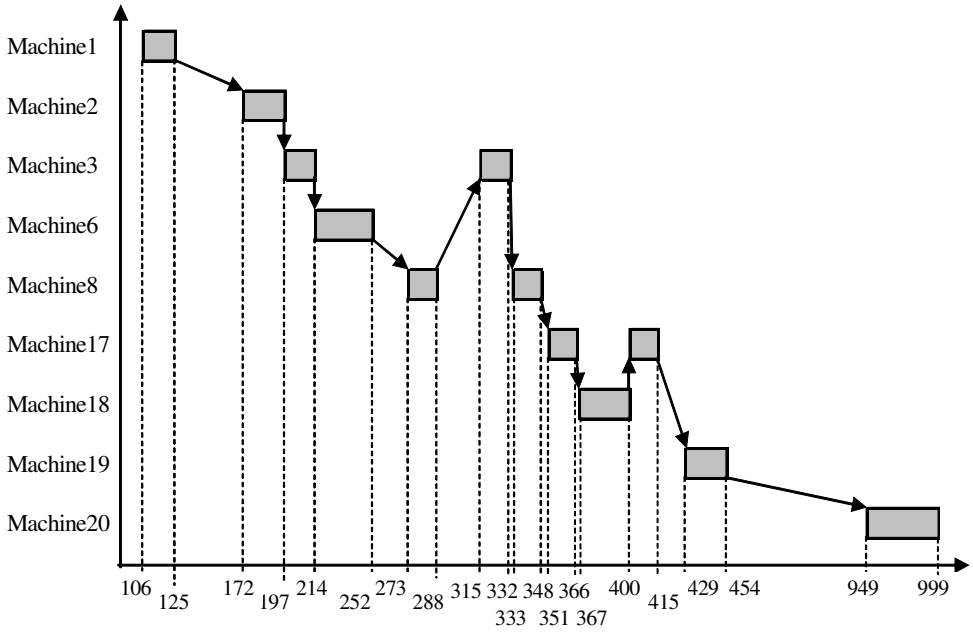


Figure 7.9 The Gannt diagram of a job

Table 7.10 The start and finish times of the jobs for machine 3

job	machine	processing time	start time	finish time
1	3	28	163	191
6	3	17	197	214
3	3	5	273	278
4	3	4	278	282
9	3	33	282	315
6	3	17	315	332
2	3	10	332	342
5	3	7	342	349
10	3	33	349	382
7	3	6	382	388
8	3	7	388	395
11	3	18	395	413
11	3	18	452	470
12	3	11	470	481
13	3	11	481	492
17	3	24	492	516
20	3	17	516	533
18	3	22	533	555
18	3	22	604	626
20	3	17	673	690
19	3	11	690	701
19	3	11	724	735

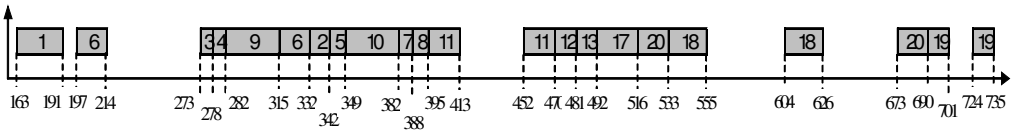


Figure 7.10 The Gannt diagram of machine3

In the remainder of this section we compare the results of the MSBH with the FCFS dispatching rule. The results are given in Table 7.11. Analysis of Table 7.11 reveals that the MSBH outperforms the FCFS rule in all of the 28 days based on solution quality. The FCFS rule finds the results in 71.06% less time than the MSBH. The average solution gap between the MSBH and the FCFS rule is 9.52%. The maximum solution gap is 17.82%, the minimum solution gap is 5.24%. Table 7.12 gives the total procrastinated minutes for each machine for the FCFS rule. The average time for machine 20 is 696.68 minutes which is 525 minutes for the MSBH as it is seen in Table 7.4.

We compared the results to evaluate the differences using the paired t-test at the 1% level of significance. On both solution time and quality, the difference between the MSBH and the FCFS rule is statistically significant.

Table 7.9 The results of FCFS rule

	MSBH		**FCFS**		**FCFS - MSBH**			
					C_{max}		**time**	
day	**C_{max}**	**time**	**C_{max}**	**time**	**ratio**	**%RE**	**ratio**	**%RE**
1	1798	491	2112	137	1.175	17.47	0.279	-72.10
2	2121	1013	2436	427	1.148	14.84	0.422	-57.85
3	1768	544	2083	180	1.178	17.82	0.331	-66.91
4	2000	376	2315	96	1.158	15.78	0.255	-74.47
5	1790	133	2105	28	1.176	17.63	0.211	-78.95
6	1425	26	1662	3	1.167	16.66	0.115	-88.46
7	1751	164	1909	47	1.091	9.05	0.287	-71.34
8	2144	618	2303	180	1.074	7.43	0.291	-70.87
9	2447	765	2607	261	1.065	6.53	0.341	-65.88
10	2042	185	2203	74	1.079	7.88	0.400	-60.00
11	1780	330	1942	75	1.091	9.08	0.227	-77.27
12	2229	410	2391	99	1.073	7.28	0.241	-75.85
13	1679	113	1841	23	1.097	9.66	0.204	-79.65
14	2195	296	2358	77	1.074	7.41	0.260	-73.99
15	1927	330	2090	79	1.085	8.45	0.239	-76.06
16	1670	99	1834	22	1.098	9.79	0.222	-77.78
17	1326	59	1462	6	1.102	10.22	0.102	-89.83
18	1307	29	1393	4	1.065	6.55	0.138	-86.21
19	1528	56	1675	11	1.096	9.64	0.196	-80.36
20	1560	75	1693	21	1.086	8.56	0.280	-72.00
21	1670	271	1804	66	1.081	8.06	0.244	-75.65
22	1850	271	1984	116	1.073	7.27	0.428	-57.20
23	2365	114	2500	55	1.057	5.71	0.482	-51.75
24	2593	134	2729	65	1.052	5.24	0.485	-51.49
25	2583	372	2720	198	1.053	5.30	0.532	-46.77
26	2573	123	2711	58	1.054	5.36	0.472	-52.85
27	2445	86	2583	18	1.056	5.64	0.209	-79.07
28	2196	24	2335	5	1.063	6.33	0.208	-79.17
avg	**1956**	**268**	**2135**	**87**	**1.095**	**9.52**	**0.289**	**-71.06**

Table 7.12 The total procrastinated minutes for each machine for the FCFS rule

	machine																			
day	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	672
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	996
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	643
4	0	0	0	0	0	0	0	0	0	0	0	0	48	60	0	0	85	0	127	875
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	665
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	222
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	469
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	863
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1167
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	763
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	502
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	951
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	401
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	90	918
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	650
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	394
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	235
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	253
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	364
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	544
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	111	140	1060
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1289
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1280
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1271
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1143
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	895
avg	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**1.71**	**2.14**	**0.00**	**0.00**	**3.04**	**5.93**	**13.46**	**696.68**

When we compare the results of the MSBH and the FCFS dispatching rule, with the MSBH, the average completion times of the jobs and average waiting times of the jobs in the shop decrease, and machine utilization rates increase. Because the makespan values of the FCFS rule for each day are higher than the MSBH, the machine utilization rates of the MSBH are high. Table 7.13 shows the average completion times of the jobs for the MSBH and the FCFS rule. Analysis of the table reveals that, with the MSBH, the jobs are completed in shorter times than with the FCFS rule. The average completion time is 929.92 minutes for the MSBH, whereas it is 1187.28 minutes for the FCFS rule, and it is 28% more than the MSBH.

Table 7.13 The average completion times of the jobs for the MSBH

day	MSBH	FCFS
1	331.61	783.98
2	908.63	1209.18
3	1017.71	1368.27
4	727.93	1149.93
5	951.22	1285.07
6	808.00	1045.60
7	416.53	754.32
8	827.13	1012.45
9	1276.48	1464.45
10	1343.54	1532.59
11	948.37	1143.26
12	753.86	1103.66
13	1056.45	1244.40
14	653.34	1025.07
15	1146.64	1332.00
16	840.35	1029.85
17	649.83	822.00
18	724.30	818.50
19	680.60	880.60
20	502.73	812.00
21	460.02	763.83
22	676.07	901.12
23	555.33	1216.76
24	1509.35	1662.35
25	1676.69	1834.72
26	1575.12	1732.94
27	1575.15	1731.80
28	1444.92	1583.25
avg	**929.92**	**1187.28**

CHAPTER EIGHT
CONCLUSIONS

The reentrant job shop scheduling problem (RJSSP) is important because the assumption that each job visits each machine only once is often violated in practice. The makespan objective function has been rarely studied in literature for the RJSSPs. The problem of minimizing makespan in a reentrant job shop is theoretically challenging. Modified versions of the shifting bottleneck procedure (SBH) with total weighted tardiness, maximum lateness performance measures are widely used in the literature to solve various types of RJSSPs. In this study, a modified SBH for the RJSSP with makespan minimization objective has been proposed and applied to several test problems as well as the application problem itself.

We have evaluated the performance of the MSBH through extensive computational experiments on an extensive set of randomly generated test problems of reentrant job shops. The computational experiments show that the MSBH is effective for the RJSSP. As a matter of fact, as the size of the problems becomes larger and larger in practice, our proposed method seems able to meet the practical demands for both a quite good quality of solutions and a moderate computing time. The computational burden of the original SBH increases rapidly as the number of operations increases and the method is probably impractical for problems with more than about 500 operations.

Subproblem prioritization procedures play an important role in the MSBH. If the MSBH can recognize the importance of the machines and always schedule the critical machines first, the performance could be further improved. When the bottleneck machines are scheduled well, the schedule of other non-bottleneck machines will easily fall into place. Hence, the MSBH has less need to reschedule, reducing both the number of subproblems solved and the solution time. In this study, Total Machine Load (TML) and Total Infeasibility (TOTINF) subproblem criticality measures have been also adapted to the MSBH.

Subproblem solution procedures have a significant effect on both solution quality and computation time. A poor subproblem solution may lead to choosing the wrong machine as bottleneck early in the solution process, which will lead to a poor overall solution later on. Furthermore, a poor subproblem solution procedure will limit the usefulness of the resequencing step. We prove that our modified algorithm for solving the single machine maximum lateness (L_{max}) problem is effective especially for large instances.

The result of the experiments shows that the performance of the MSBH depends on the amount of resequencing. Resequencing step has a significant effect on both solution quality and computation time. It increases the solution quality as well as the solution time. Therefore, the trade-off between solution time and quality is an important point.

In order to reduce the number of operations and machines in the MSBH, a modified rolling horizon procedure (MHRP) has been proposed to decompose large reentrant job shops with makespan minimization objective. The MHRP has been tested for different number of time windows and various rules for assignment of operations to time windows. Furthermore, a new rule has been developed for the special RJSSP concerning the application problem. The MSBH and its two versions MSBH_TML and MSBH_TOTINF consistently outperform the MRHP on solution quality. In general, the MSBH obtained the best solutions among the other proposed methods, but it spent more time than the MRHP for large problems.

Finally, it must be point outed that, there are different job shop environments that contain batching tool groups, parallel machines, sequence dependent setup times. Hence another extension of this study can be to modify the MSBH for these types of reentrant job shops for makespan minimization objective. Although the other performance measures such as total weighted tardiness, maximum lateness etc. can be analyzed.

REFERENCES

Aarts, E., & Lenstra, J. K. (1997). *Local search combinatorial optimization.* New York: Wiley.

Aarts, E. H. L., Van Laarhoven, P. J. M., Lenstra, J. K. & Ulder, N. L. J. (1994). A computational study of local search algorithms for job shop scheduling. *ORSA Journal on Computing*, 6, 118–125.

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science,* 34(3), 391-401.

Akcora, E., Gramsan, S. E., & Saygın, C. (2005). A job shop scheduling heuristic for varying reward structures. *Mathematical and Computer Modelling*, 42(11-12), 1163-1174.

Akpan, E. O. P. (1996). Job shop sequencing problems via network scheduling technique. *International Journal of Operations and Production Management*, 16(3), 76-86.

Al-Anzi, F. S., & Allahverdi, A. (2006). A hybrid tabu search heuristic for the two-stage assembly scheduling problem. *International Journal of Operations Research*, 3(2), 109-119.

Alvarez-Valdes, R., Fuertes, A., Tamarit, J. M., Gimenez, G., & Ramos, R. (2005). A heuristic to schedule flexible job-shop in a glass factory. *European Journal of Operational Research*, 165, 525-534.

Anderson, E. J, & Nyirenda, J. C. (1990). Two new rules to minimize tardiness in a job shop. *International Journal of Production Research*, 28(12), 2277-2292.

Applegate, D., & Cook, W. (1991). A computational study of job shop scheduling. *ORSA Journal of Computing*, 3, 149–156.

Ashour, S., Chiu, K., & Moore, T. (1973). An optimal schedule time for a job shop-like disjunctive graph. *Networks*, 3, 333-349.

Ashour, S., & Parker, R. (1971). A precedence graph algorithm fort he shop scheduling problem. *Operational Research Quarterly*, 22, 165-175.

Aydın, M.E., & Öztemel, E. (2000). Dynamic job-shop scheduling using reinforcement learning agents. *Robotics and Autonomous Systems*, 33, 169-178.

Aytug, H., Kempf, K., & Uzsoy, R. (2002). Measures of subproblem criticality in decomposition algorithms for shop scheduling. *International Journal Of Production Research*, 41(5), 865-882.

Baker, K.R. (1974). *Introduction to sequencing and scheduling*. New York: John Wiley.

Baker, K. R., & Su, Z. S. (1974). Sequencing with due dates and early start times to minimize maximum tardiness. *Naval Research Logistics Quarterly*, 21, 171-176.

Balas, E. (1969). Machine sequencing via disjunctive graphs: an implicit enumeration approach. *Operations Research*, 17, 941-957.

Balas, E., Lancia, G., Serafini, P., & Vazacapulos, A. (1998). Job shop scheduling with deadlines. *Journal of Combinatorial Optimization*, 1, 329-353.

Balas, E., Lenstra, J. K., & Vazacopoulos, A. (1995). The one machine scheduling with delayed precedence constraints. *Management Science*, 41, 94-109.

Balas, E., & Vazacopoulos, A. (1998). Guided local search with shifting bottleneck for job shop scheduling. *Management Science*, 44, 262-275.

Baptiste, P., Flamini, M., & Sourd, F. (in press). Lagrangian bounds for just-in-time job-shop scheduling. *Computers & Operations Research*.

Barnes, J. W. & Chambers, J. B. (1995). Solving the job shops cheduling problem with tabu search. *IIE Transactions*, 27, 257-263.

Bhaskaran, K., & Pinedo, M. (1991). *Dispatching*. In: Handbook of Industrial Engineering. G. Salvendy (ed.), New York: Wiley.

Blackstone, J., Phillips, D., & Hogg, G. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research*, 20(1), 27-45.

Blazewicz, J., Domschke, W., & Pesch, E. (1996). The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research*, 93, 1-33.

Blazewicz, J., Dror, M., & Weglarz, J. (1991). Mathematical programming formulations for machine scheduling: a survey. *European Journal of Operational Research*, 51, 283-300.

Blazewicz, J., Pesch, E., & Sterna, M. (2000). The disjunctive graph machine representation of the job shop scheduling problem. *European Journal of Operational Research*, 127, 317-331.

Bowman, E. H. (1959). The scheduling-sequence problem. *Operations Research*, 7, 621-624.

Brucker, P., Jurisch, B., & Sievers, B. (1994). A branch and bound algorithm for the job shop scheduling problem. *Discrete Applied Mathematics*, 49, 102-127.

Carlier, J., & Pinson, E. (1989). An algorithm for solving the job-shop problem. *Management Science*, 35, 164-176.

Cavalieri, F. S., Crisafulli, O., & Mirabella, O. (1999). A genetic algorithm for job shop scheduling in a semiconductor manufacturing system. *IEEE Transactions of Engineering Management*, 41(2), 957–961.

Cavory, G., Dupas, R., & Goncalves, G. (2005). A genetic approach to solving the problem of cyclic job shop scheduling with linear constraints: IEPM: Focus on scheduling. *European Journal of Operational Research*, 161(1), 73-85.

Chandrasekaran, M., Asokan, P., Kumanan, S., Balamurugan, T., & Nickolas, S. (2006). Solving job shop scheduling problems using artificial immune system. *International Journal of Manufacturing Techniques*, 31, 580-593.

Charlton J., Death, C. (1969). A general method for machine scheduling. *Operational Research Quarterly*, 21, 127-134.

Chung, D., Lee, K., Shin, K., & Park, J. (2005). A new approach to job shop scheduling problems with due date constraints considering operation subcontracts. *International Journal of Production Economics*, 98(2), 238-250.

Conway, R. W., Maxwell W. L., & Miller L. W. (1967). *Theory of scheduling*. England MA: Addison-Wesley.

Croce, F. D., Tadei, R., & Volta, G. (1995). A genetic algorithm for the job shop problem. *Computers & Operations Research*, 22, 15-24.

Dabbas, R. M., Chen, H.-N., Fowler, J. W., & Shunk, D. L. (2001). A combined dispatching approach to scheduling semiconductor manufacturing systems. *Computers and Industrial Engineering*, 39, 307-324.

Dauzere-Peres, S., & Lasserre, J. B. (1993). A modified shifting bottleneck procedure for job shop scheduling. *International Journal of Production Research,* 31, 923-932.

Dell'Amico, M., & Trubian, M. (1993). Applying tabu search to the job shop scheduling problem. *Annals of Operations Research*, 41, 231-252.

Demirkol, E., Mehta, S., & Uzsoy, R. (1997). A computational study of shifting bottleneck procedures for shop scheduling. *Journal of Heuristics*, 3(2), 111-137.

Demirkol, E., & Uzsoy, R. (1997). Performance of decomposition methods for complex workshops under multiple criteria. *Computers Industrial Engineering*, 33(1-2), 261-264.

Demirkol, E., Uzsoy, R., Ovacik, I. M. (1995). Decomposition algorithms for scheduling semiconductor testing facilities. *Electronics Manufacturing Technology Symposium, 1995, 'Manufacturing Technologies - Present and Future', Seventeenth IEEE/CPMT International*, 199-204.

Dobson, G. & Karmakar, N.S. (1986). *Large scale shop scheduling: formulations and decomposition. Graduate School of Management Working Paper, Series No.QM8631, University of Rochester.*

Dorndorf, U., & Pesch, E. (1993). Genetic algorithms for job shop scheduling. *Operations Research Proceedings, 1992, Springer, Berlin*, 243-250.

Dorndorf, U., & Pesch, E. (1996). Evolution-based learning in a job shop scheduling environment. *Computers & Operations Research*, 22, 25-40.

Emre, A. V. (2001). Computer simulation of due-date setting in multi-machine job shops. *Computers and Industrial Engineering*, 41, 77-94.

Fisher, M. L. (1973). Optimal solution of scheduling problems using Lagrange multipliers: Part 1. *Operations Research*, 21, 1114-1127.

Fisher, M. L., Lageweg, B. J., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1983). Surrogate duality relaxation for job shop scheduling. *Discrete Applied Mathematics*, 5, 65-70.

French, S. (1982). *Sequencing and scheduling: An introduction to the mathematics of the job shop*. New York: John-Wiley.

Garey, M. R., & Johnson, D. S. (1979). *Computer and intractability: A guide to the theory of NP-completeness*. San Francisco,CA: W. H.Freeman.

Giffler, B., & Thompson, G. L. (1960). Algorithms for solving production scheduling problems. *Operations Research*, 8, 487-503.

Gonçalves, J. F., Mendes, J. J. M., & Resende, M. G. C. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, 167(1), 77-95.

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5, 287–326.

Guinet, A. (2000). Efficiency of reductions of job shop to flow shop problems. European Journal of Operational Research, 125(3), 469-485.

Guo, Y., Hoede, C., & Smit, G. J. M. (2006). A pattern selection algorithm for multi-pattern scheduling. *In: Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - 12th Reconfigurable Architecture Workshop (RAW 2006), 25-29 Apr 2006, Rhodes Island, Greece. pp. 198-205. IEEE Computer Society. ISBN 1-4244-0054-6*

Gupta, A. K., & Sivaskumar, A. I. (2006). Job shop scheduling techniques in semiconductor manufacturing. *International Journal of Advanced Manufacturing Technologies*, 27, 1163-1169.

Haupt, R. (1989). A survey of priority-rule based scheduling. *OR Spektrum*, 11, 3-16.

Ho, N. B., Tay, J. C., & Laı, E. M-K. (2007). An effective architecture for learning and evolving flexible job-shop schedules. *European Journal of Operations Research*, 179 (2), 316-333.

Hoitomt, D. J., Luh, P. B., & Pattipati, K. R. (1993). A practical approach to job shop scheduling problems. *IEEE Transactions on Robotics and Automation*, 9, 1-13.

Holthaus, O., & Rajendran, C. (1997). Efficient Dispatching Rules for Scheduling in a job shop. *International Journal of Production Economics*, 48(1), 87-105.

Holtsclaw, H. H., & Uzsoy, R. (1996). Machine criticality measures and subproblem solution procedures in shifting bottleneck methods: a computational study. *Journal of Operational Research Society*, 47(5), 666-677.

Hopp, W. J., Spearman, M. L., Chayet, S., Donohue, K. L., & Gel, E. S. (2002). Using an optimized queueing network model to support wafer fab design. *IIE Transactions*, 34(2), 119-130.

Huang, K. L., & Liao, C. J. (2004). Ant colony optimization combined with taboo search for the job shop scheduling problem. *IEEE Transactions on Automation Science and Engineering*, 1, 1-10.

Ivens, P., & Lambrecht, M. (1996). Extending the shifting bottleneck procedure to real-life applications. *European Journal of Operational Research*, 90(2), 252-268.

Jain, A. S., & Meeran, S. (1999). Deterministic job shop scheduling: past, present and future. *European Journal Operations Research*, 113, 390-434.

Jayamohan, M. S., & Rajendran, C. (2004). Development and analysis of cost-based dispatching rules for job shop scheduling. *European Journal of Operational Research*, 157(2), 307-321.

Jeong, H., Park, J. & Leachman, R.C. (1999). A batch splitting method for a job shop scheduling problem in an MRP environment. *International Journal of Production Research*, 37(15), 3583-3598.

Johnson, S. M. (1954). Optimal two and three stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1, 61-67.

Kaschel, J., Teich, T., & Zacher, B. (2002). Real-time dynamic shop floor scheduling using Evolutionary Algorithms. *International Journal of Production Economics*, 79(2), 113-120.

Kaskavelis, C., & Caramanis, M. (1994). Application of a lagrangian relaxation based scheduling algorithm to a semiconductor testing facility. *Proc. Fourth International Conf. on CIM and Automation Technology, Troy, NY, October*, 106-112.

Kaskavelis, C., & Caramanis, M. (1998). Efficient lagrangian relaxation algorithms for industry size job-shop scheduling problems. *IIE Transactions*, 30(11).

Kubiak, W., Lou, S. X. C., & Wang, Y. (1996). Mean flow time minimization in reentrant job shops with a hub. *Operations Research*, 44, 764-776.

Kumar, S., & Kumar, P. R. (2001). Queueing network models in the design and analysis of semiconductor wafer fabs. *IEEE Transactions on Robotics and Automation*, 17(5), 548-561.

Kuroda, M., Tomita, T., & Maeda, K. (1999). Dynamic control of a cellular-line production system under variations in the product mix. *International Journal of Production Economics*, 60-61, 439-445.

Kuroda, M., & Wang, Z. (1996). Fuzzy job shop scheduling. *International Journal of Production Economics*, 44, 45-51.

Lageweg, B., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1977). Job shop scheduling by implicit enumeration. *Management Science*, 24, 441-450.

Lam, F. S. C., Lin, B. C., Sriskandarajah, C., Yan, H. (1999). Scheduling to minimize product design time using a genetic algorithm. *International Journal Of Production Research*, 37(6), 1369–1386.

Lambrecht, M. R., Ivens P., & Vandaele, N. J. (1998). ACLIPS: A capacity and lead time integrated procedure for scheduling. *Management Science*, 44, 1548-1561.

Lasserre, J. B. (1992). An integrated model for job shop planning and scheduling. *Management Science*, 38, 1201-1211.

Lawrence, S. (1984). Supplement to resource constraint project scheduling: An experimental investigation of heuristic scheduling techniques. *Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, USA, October.*

Lawrence, S. R., & Sewell, E. C. (1997). Heuristic, optimal, static, and dynamic schedules when processing times are uncertain. *Journal of Operations Management*, 15(1), 71-82.

Lee, Y.H., Bhaskaran, K., & Pinedo, M.L. (1997). A heuristic to minimize the total weighted tardiness with sequence dependent setups. *IIEE Transactions*, 29, 45-52.

Lengley, A., Hatono, I., & Ueda, K. (2003). Scheduling for on-time completion in job shops using feasibility function. *Science Direct Computers & Industrial Engineering*, 45, 215-229.

Lenstra, J. K., Rinnooy Kan, A. H .G., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals Discrete Mathematics*, 1, 343-362.

Liao, C.-J., & You, C.-T. (1992). An improved formulation for the job-shop scheduling problem. *Journal of Operational Research Society*, 23(11), 1047-1054.

Lourenco, H. R. (1995). Job shop scheduling: Computational study or local search and large-step optimization methods. *European Journal Of Operational Research*, 83, 347-364.

Luh, P. B., & Hoitomt, D. J. (1993). Scheduling of manufacturing system using the Lagrangian relaxation technique. *IEEE Transactions on automatic control, special issue: Meeting the challenge of computer science in the industrial applications of control*, 38, 1066-1080.

Manne, A. S. (1960). On the job shop scheduling problem. *Operations Research*, 8, 219-223.

Martin, P. D., & Shmoys, D. B. (1996). A new approach to computing optimal schedules for the job shop scheduling problem. *S.T. McCormick W.H. Curnigham, M. Queyranne, eds., Proceedings fifth international IPCO conference, Vancouver, Canada. LNCS 1084*, 389–403.

Mason, S. J., Fowler J. W., & Carlyle, W. M. (2002). A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shop. *Journal of Scheduling*, 5(3), 247-262.

Mason, S. J., Fowler, J. W., Carlyle, W. M., & Montgomery, D. C. (2005). Heuristics for minimizing total weighted tardiness in complex job shops. *International Journal of Production Research*, 43(10), 1943-1963.

Mason, S. J., & Oey, K. (2003) Scheduling complex job shops using disjunctive graphs :A cycle elimination procedure. *International Production Research*, 41(5), 981-994.

Matsuo, H., Suh, C. J., & Sullivan, R. S. (1988). A controlled search simulated annealing method for the general job shop scheduling problem. *Working paper, 03-44-88, Graduate School of Business, University of Texas, Austin*.

Mcmahon, G.B., & Florian, M. (1975). On scheduling for ready times and due dates to minimize maximum lateness. *Operations Research*, 23, 475-482.

Miragliotta, G., & Perona, M. (2005). Decentralised, multi-objective driven scheduling for reentrant shops: A conceptual development and a test case. *European Journal Of Operational Research*, 167(3), 644-662.

Morten, T. E., & Pentico, D. W. (1993). *Heuristic scheduling systems*. New York: Wiley.

Mönch, L., & Driebel, R. (2005). A distributed shifting bottleneck heuristic for complex job shops. *Computers & Industrial Engineering*, 49, 363-380.

Mönch, L., Schabacker, R., Pabst, D., & Fowler, J.W. (2007). Genetic algorithm-based subproblem solution procedures for a modified shifting bottleneck heuristic for complex job shops. *European Journal of Operational Research*, 177, 2100-2118.

Mönch, L., Rose, O., & Sturm, R. (2003). A simulation framework for performance assessment of shop-floor control systems. *Simulation: Transactions of the Society of Modeling and Computer Simulation International*, 79(3), 163-170.

Muth, J. P., & Thompson, G. L. (1963). *Industrial scheduling*. Englewood Cliffs, NJ: Prentice-Hall.

Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and combinatorial optimization* New York: J. Wiley.

Nose, K., Hiramatsu, A., & Konishi, M. (1999). Using genetic algorithm for job shop scheduling problems with reentrant product flows. *0-7803-5670-5/99/IEEE*.

Nowicki, E., & Smutnicki, C. (1996). A fast taboo search algorithm for the job shop problem. *Management Science*, 42(6), 797-813.

Odrey, N. G., Gren, J. D., & Appello, A. (2001). A generalized Petri net modeling approach for the control of re-entrant flow semiconductor wafer fabrication. *Robotics and Computer-Integrated Manufacturing*, 17(1-2), 5-11.

Oey, K., & Mason, S. J. (2001). Scheduling batch processing machines in complex job shops. *Proceedings of the 2001 Winter Simulation Conference*, 1200-1207.

Ohta, H., & Nakatani, T. (2006). A heuristic job-shop scheduling algorithm to minimize the total holding cost of completed and in-process products subject to no tardy jobs. *International Journal of Production Economics*, 101(1), 19-29.

Ovacik, I. M., & Uzsoy, R. (1992). A shifting bottleneck algorithm for scheduling semiconductor testing operations. *Journal Of Electronic Manufacturing*, 2, 119-134.

Ovacık, I. M., & Uzsoy, R. (1997). *Decomposition methods in large scale job shops*. Baston: Kluwer Academic Press.

Pabst, D. (2003). Handling precedence constraints for the shifting bottleneck heuristic applied in a dynamic semiconductor manufacturing environment. *M.S. thesis. Würzburg: Universitat Würzburg.*

Pan, J. C.-H. (1997). A study of integer programming formulations for scheduling problems. *International Journal Of Systems Science*, 28, 33-41.

Pan, . J. C.-H., & Chen J.-S. (2005). Mixed binary integer programming formulations for the reentrant job shop scheduling problem. *Computers & Operations Research*, 32, 1197-1212.

Panwalkar, S.S., & Iskander, W. (1977). A survey of scheduling rules. *Operations Research*, 25, 45-61.

Petrovic, S., Fayad, C., Petrovic, D., Burke, E., & Kendall, G. (2006). Fuzzy job shop scheduling with lot-sizing. *Anals of Operations Research*, in press.

Pezzella, F., & Merelli, E. (2000). Tabu search method guided by shifting bottleneck for the job shop scheduling problem. *European Journal of Operational Research*, 120(2), 297-310.

Pfund, M., Yu, L., Fowler, J., & Carlyle, W. (2002). The effects of processing time variability and equipment downtimes on various scheduling approaches for a printed wiring board assembly operation. *Journal of Electronics Manufacturing*, 11(1), 19-31.

Pinedo, M., & Chao, X. (1999). *Operations scheduling with applications in manufacturing and services*. Boston: Irwin/McGraw-Hill.

Pinedo, M., & Singer, M. (1996). A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Research Report, Department of Industrial Engineering and Operations Research, Columbia University.*

Pinedo, M., & Singer, M. (1999). A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Naval Research Logistics*, 46, 1-17.

Rahmati, A. (1998). Representation of hybrid genetic algorithm for solving non-classic job shop scheduling problems. *MS Eng. Thesis. Faculty of Engineering, University of Tehran (in Persian).*

Raman, N., & Talbot, F. B. (1993). The job shop tardiness problem: a decomposition approach. *European Journal of Operational Research*, 69, 187-199.

Ramudhin, A., & Marier, P. (1996). Generalized shifting bottleneck procedure. *European Journal of Operational Research*, 93(1), 34-48.

Rinnooy Kan, A. H. G. (1976). *Machine scheduling problems: Classification, complexity, and computations*. Nijhoff: The Hague.

Roobeek, F. (1997). Better choice –a fuzzy logic based lot sequencing decision support system for operators in a job shop fab with reentrant process flows. *Proceedings of the 1997 IEEE International Symposium on Semi-conductor on Manufacturing Conference*.

Sadeh, N. (1991). Look-ahead techniques for micro-opportunistic job shop scheduling. *PhD Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA*.

Satake, T., Morikawa, K., Takahashi, K., & Nakamura, N. (1999). Simulated annealing approach for minimizing the makespan of the general job shop. *International Journal of Production Economics*, 60, 515-522.

Schutten, M. (1998). Practical job shop scheduling. *Annals of Operational Research*, 83, 161-177.

Shafaei R., & Brunn, P. (1999). Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. *International Journal of Production Research*, 37(17), 3913-3925.

Singer, M. (2001). Decomposition methods for large job shops. *Computer & Operations Research*, 28, 193-207.

Singer, M., & Pinedo, M. (1998). A computational study of branch and bound techniques for minimizing the total weighted tardiness in job shops. *IIE Transactions Scheduling and Logistics*, 30, 109-118.

Spachis, A. S., & King, J. R. (1979). Job shop scheduling heuristics with local neighbourhood search. *International Journal of Production Research*, 17(6), 507-526.

Steinhöfel, K., Albrecht, A., & Wong, C.K. (1999). Two simulated annealing-based heuristics for the job shop scheduling problem. *European Journal of Operational Research*, 118(3), 524-548.

Storer, R. H., Wu, S. D., & Vaccari, R. (1992). New search spaces for sequencing problems with application to job shop scheduling. *Management Science*, 38, 1495-1509.

Storer, R. H., Wu, S. D., & Vaccari, R. (1995). Problem and Heuristic Space Search Strategies for Job Shop Scheduling. *ORSA Journal on Computing*, 7, 453-467.

Su, L. H., Chang, P. C., & Lee, E. S. (1998). A heuristic for scheduling general job shops to minimize maximum lateness. *Mathematical Computing Modeling*, 27(1), 1-15.

Sun, X., & Noble, J. S. (1999). An approach to job shop scheduling with sequence-dependent setups. *Journal of Manufacturing Systems*, 18, 416-430.

Taillard, E. (1994). Parallel taboo search techniques for the job shop scheduling problem. *ORSA Journal of Computing*, 6, 108-117.

Tavakkoli-Moghaddam, R., & Daneshmand-Mehr, M. (2005). A computer simulation model for job shop scheduling problems minimizing makespan. In press.

Uzsoy, R., Church, L. K., Ovacik, I., & Hinchman, J. (1993). Performance evaluation of dispatching rules for semiconductor testing operations. *Journal of Electronics Manufacturing*, 3, 95-105.

Uzsoy, R., Lee, C. Y., & Martin-Vega, L. A. (1992). A review of production planning and scheduling models in the semiconductor industry, part 1: system characteristics, performance evaluation and production planning. *IEE Transactions*, 24, 47-61.

Uzsoy, R., Martin-Vega, L. A., Lee, C.-Y., & Leonard, P. A. (1991). Production scheduling algorithms for a semiconductor test facility. *IEEE Transactions on Semiconductor Manufacturing*, 4, 270-280.

Uzsoy, R., & Wang, C. S. (2000). Performance of decomposition algorithms for job shop scheduling problems with bottleneck machines. *International Journal of Production Research*, 38, 1271-1286.

Vaessens, R. J. M., Aarts, E. H. L., & Lenstra, J. K. (1996). Job shop scheduling by local search. *INFORMS Journal on Computing*, 8, 302-317.

Van Laarhoven, P. J. M., Aarts, E. H. L., & Lenstra, J. K. (1992). Job shop scheduling by simulated annealing. *Operations Research*, 40, 113-125.

Vargas-Villamil, F. D., & Rivera, D. E. (2001). A model predictive control approach for real-time optimization of reentrant manufacturing lines. *Computers in Industry*, 45, 45-57.

Vargas-Villamil, F. D., Rivera, D. E., & Kempf, K. G. (2003). A hierarchical approach to production control of reentrant semiconductor manufacturing lines. *IEEE Transactions on Control Systems Technology*, 11(3), 578-587.

Vepsalainen, A. P. J., & Morton, T. E. (1987). Priority rules and lead time estimation for job shop scheduling with weighted tardiness costs. *Management Science*, 33, 1037-1047.

Vepsalainen, A. P. J., and Morton, T. E. (1988). Improving local priority rules global lead-time estimates: a simulation study. *Journal of Manufacturing and Operations Management*, 1, 102-118.

Wagner, H. M. (1959). An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6, 131-140.

Watanabe, M., Ida, K., & Gen, M. (2005). A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem. *Computers & Industrial Engineering*, 48(4), 743-752.

Wenqi, H., & Aihua, Y. (2004). An improved shifting bottleneck procedure for the job shop scheduling problem. *Computers & Operations Research*, 31, 2093-2110.

White, K. P., & Rogers, R. V. (1990). Job shop scheduling: Limits of the binary disjunctive formulation. *International Journal of Production Research*, 28, 2187-2200.

.

Wilson, J. M. (1989). Alternative formulations of a flow-shop scheduling problem. *Journal of The Operational Research Society*, 40, 395-399.

Wu, C.-S., Li, D.-C., & Tsai, T.-I. (2006). Applying the fuzzy ranking method to the shifting bottleneck procedure to solve scheduling problems of uncertainty. *The International Journal of Advanced Manufacturing Technology,* 31(1-2), 98-106.

Yeung, T., & Mason, S. J. (2006). Using real analysis to value reoptimization options in the shifting bottleneck heuristic. *Naval Research Logistics*, 53, 285-297.

Yoo, W. S., & Martin Vega, L. A. (1997). A decomposition methodology for scheduling semiconductor test operations for number of tardy job measures. *Journal of Electronic Manufacturing*, 7, 51-60.

Yu, H., & Liang, W. (2001). Neural network and genetic algorithm-based hybrid approach to expanded job shop scheduling. *Computers and Industrial Engineering*, 39(3/4), 337-356.

Zhang, C., Li, P. , Rao, Y., & Li, S. (2005). A new hybrid GA/SA algorithm for the job shop scheduling problem. *Lecture Notes in Computer Science Springer Berlin/Heidelber Evolutionary Computation in Combinatorial Optimization*, 3448, 246-259.

Zhang, D., Li, T., & Li, S. (2005). An improved shifting bottleneck algorithm for job shop scheduling problem. *Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design 2005*, 1112-1116.

Zhang, H. C., & Huang, S. H. (1995). Applications of neural networks in manufacturing: a state-of-the-art survey. *International Journal of Production Research*, 33(3), 705–728.

Zhogby, J., Barnes, W., & Hasenbein, J. J. (2005). Modeling the reentrant job shop scheduling problem with setups for metaheuristic searches. *European Journal of Operational Research*, 167, 336-348.

APPENDIX A

The MATLAB Code Of The Modified Shifting Bottleneck Heuristic

```
clear; clc;
load machine
tcpu=cputime;
reschlevel=6;
Bottleneck=0; job=0; jb=0; time=0; tm=0; kume=0;
Lmax=0;Lmaxx=0;Lmaxxx=0;Lmaxxxxx=0; jx=0;
tes=0; tef=0; tgf=0; tef=0;
itno=1;
for i=1:a
    tes(i)=0;
    tef(i)=0;
end;
for i=1:a
   if pr1(i)==0
      tes(i)=0;
      tef(i)=t(i);
   elseif pr1(i)~=0 & pr1(i)<i & tef(pr1(i))~=0
       tef(i)=t(i)+tef(pr1(i));
       tes(i)=tef(pr1(i));
end;
end;
 for i=1:a
    tgs(i)=0;
    tgf(i)=0;
end;
cmax=max(tef);
for i=a:-1:1
   if sc1(i)==0
      tgf(i)=cmax;
      tgs(i)=tgf(i)-t(i);
  elseif sc1(i)>i & tgf(sc1(i))~=0
       tgs(i)=tgs(sc1(i))-t(i);
       tgf(i)=tgs(sc1(i));
     end;
end;
for i=1:a
  table1r0(i,1)=i;  table1r0(i,2)=t(i);
table1r0(i,3)=tes(i);  table1r0(i,4)=tgf(i);
end;
for j=1:m
  vv=0;
       for i=1:a
           if k(i)==j

vv=vv+1;tt(j,vv)=table1r0(i,3)+table1r0(i,2);
           end;
            v(j)=vv;
      end;
   if v(j)==0
      tt(j,:)=0;
   end;
end;
for j=1:m
     for z=1:v(j)
         ttt(z)= tt(j,z);
     end;
          ttmin(j)=min(ttt);
ttt=0;
end;
for j=1:m
  uu=0;
    for i=1:a
        if   k(i)==j & table1r0(i,3)<ttmin(j)
          uu=uu+1;jb(j,uu)=i;
        end;
      end;
      u(j)=uu;
end;
for j=1:m
  if u(j)==1 & jb(j,1)~=0
          tm(j,1)=
table1r0(jb(j,1),3)+table1r0(jb(j,1),2);
          job(j,1)=jb(j,1); time(j,1)=tm(j,1);
         fprintf('1. operation of machine %2.0f  is jb
%2.0f\n',j,job(j,1));
  elseif u(j)>1 & jb(j,1)~=0
           for h=1:u(j)
              d(j,h)=table1r0(jb(j,h),4);
                 ddd(h)= d(j,h);
           end;
           dd(j)=min(ddd);
           ddd=0;
                dddd=0;
              for g=1:u(j)
                 if table1r0(jb(j,g),4)==dd(j)
                      dddd=dddd+1;
                      r(j,dddd)= table1r0(jb(j,g),3);
                      rrr(dddd)=r(j,dddd);
                   end;
                end;
                rr(j)=min(rrr);
                rrr=0;
          for g=u(j):-1:1
              if  table1r0(jb(j,g),4)==dd(j)
                  if  table1r0(jb(j,g),3)==rr(j)
                    tm(j,1)=
table1r0(jb(j,g),3)+table1r0(jb(j,g),2);
                      job(j,1)=jb(j,g); time(j,1)=tm(j,1);
                  end;
              end;
          end;
         fprintf('1. operation of machine %2.0f  is jb
%2.0f\n',j,job(j,1));
     end;
end;
for j=1:m
if time(j,1)~=0
  for iii=2:mch(j)
     Lmaxxx(j,iii)=0;  job(j,iii)=0;  tm(j,iii)=0;
time(j,iii)=0;
   end;
    Lmaxxx(j,1)= time(j,1) - table1r0(job(j,1),4);
else
    Lmaxxx(j,:)=0;
end;
end;
j=1;
while j<=m
zz=0;yy=0;
for i=1:a
          if k(i)==j & i~=job(j,1)
             if table1r0(i,3)<=time(j,1)
                zz=zz+1; jbs(j,zz)=i; due(j,zz)=
table1r0(jbs(j,zz),4);
```

```
            elseif table1r0(i,3)>time(j,1)
                yy=yy+1; jbb(j,yy)=i;
release(j,yy)=table1r0(jbb(j,yy),3);
            end;
        end;
end;
z(j)=zz;y(j)=yy; disp('z(j) '); disp(z(j));  disp('y(j) ');
disp(y(j));
if zz>=1
fprintf ('%2.0f. machine jbs=\n', j);disp(jbs(j,:));
fprintf ('%2.0f. machine due=\n', j);disp(due(j,:));
else
jbs=0; due=0;
end;
if yy>=1
fprintf ('%2.0f. machine jbb=\n', j);disp(jbb(j,:));
fprintf ('%2.0f. machine release=\n',
j);disp(release(j,:));
else
jbb=0; release=0;
end;

for it=2:mch(j)
    if z(j)==1 & jbs(j,1)~=0
re(j)=0;  gtime=0;
for aa=1:it-1
        if    OPERATIONS(jbs(j,1)) ==
OPERATIONS(job(j,aa))
            re(j)=job(j,aa); rex(j)=aa;
        elseif re(j)==0 & OPERATIONS(jbs(j,1)) ~=
OPERATIONS(job(j,aa))
            re(j)=job(j,it-1);
        end;
end;
   if    OPERATIONS(jbs(j,1)) == OPERATIONS(re(j))
gtime = max((time(j,it-
1)+table1r0(jbs(j,1),2)),(time(j,rex(j))-
table1r0(re(j),2)+table1r0(jbs(j,1),3)-
table1r0(re(j),3)+ table1r0(jbs(j,1),2))) -
table1r0(jbs(j,1),2);
bb=0;
       if  jbb~=0
         for b=1:length(jbb(j,:))
                  if  OPERATIONS(jbb(j,b))~=
OPERATIONS(jbs(j,1))  & table1r0(jbb(j,b),3) <
gtime &  (gtime + table1r0(re(j),2)- table1r0(re(j),4))
<  (max((time(j,it-1)+table1r0(jbb(j,b),2)),
(table1r0(jbb(j,b),3)+ table1r0(jbb(j,b),2)) -
table1r0(jbb(j,b),4)))
                  bb=bb+1; gjobb(bb)= jbb(j,b);
greleaseb(bb)=table1r0(jbb(j,b),3)
            end;end;
            end;
if bb~=0
            mingrelease=min(greleaseb);
             for   bbb=bb:-1:1
                  if greleaseb(bbb)==mingrelease
                job(j,it)=gjobb(bbb); time(j,it)=
max((time(j,it-1)+table1r0(gjobb(bbb),2)),
(table1r0(gjobb(bbb),3)+ table1r0(gjobb(bbb),2)));
    end;end;
end;
if bb==0
            tm(j,it) =max((time(j,it-
1)+table1r0(jbs(j,1),2)),(time(j,rex(j))-
table1r0(re(j),2)+table1r0(jbs(j,1),3)-
table1r0(re(j),3)+ table1r0(jbs(j,1),2)));
              job(j,it) = jbs(j,1); time(j,it) =tm(j,it);
            fprintf ('%2.0f. operation of machine
%2.0f  is jb  %2.0f\n\n', it,j, job(j,it));
end;
        else
            tm(j,it) =time(j,it-1)+table1r0(jbs(j,1),2);
job(j,it) = jbs(j,1); time(j,it) =tm(j,it);
            fprintf ('%2.0f. operation of machine
%2.0f  is jb  %2.0f\n\n', it,j, job(j,it));
      end;
  Lmaxxx(j,it) = time(j,it)-table1r0(job(j,it),4);
disp('Lmaxxx='); disp(Lmaxxx);
elseif z(j)>1 & jbs(j,1)~=0
   dddd=0; r=0; rrr=0; rr=0;   d=0;          gtime=0;
            d(j)=min(due(j,:));
              for ii=1:z(j)
                  if table1r0(jbs(j,ii),4)==d(j)
                          dddd=dddd+1;
                             r(j,dddd)=
table1r0(jbs(j,ii),3);
                             rrr(dddd)=r(j,dddd);
                      end;
                  end;
                 rr(j)=min(rrr);
        for ii= z(j):-1:1
            if  table1r0(jbs(j,ii),4)==d(j) &
table1r0(jbs(j,ii),3)==rr(j)
re(j)=0;
for aa=1:it-1
        if    OPERATIONS(jbs(j,ii)) ==
OPERATIONS(job(j,aa))
            re(j)=job(j,aa); rex(j)=aa;
        elseif re(j)==0 & OPERATIONS(jbs(j,ii)) ~=
OPERATIONS(job(j,aa))
            re(j)=job(j,it-1);
        end;
end;
        if    OPERATIONS(jbs(j,ii)) ==
OPERATIONS(re(j))
gtime = max((time(j,it-
1)+table1r0(jbs(j,ii),2)),(time(j,rex(j))-
table1r0(re(j),2)+table1r0(jbs(j,ii),3)-
table1r0(re(j),3)+ table1r0(jbs(j,ii),2))) -
table1r0(jbs(j,ii),2);


bb=0;
       if  jbb~=0
 for b=1:length(jbb(j,:))
                  if OPERATIONS(jbb(j,b))~=
OPERATIONS(jbs(j,ii))  & table1r0(jbb(j,b),3) <
gtime &  (gtime + table1r0(re(j),2)- table1r0(re(j),4))
<  (max((time(j,it-1)+table1r0(jbb(j,b),2)),
(table1r0(jbb(j,b),3)+ table1r0(jbb(j,b),2)) -
table1r0(jbb(j,b),4)))
                  bb=bb+1; gjobb(bb)= jbb(j,b);
greleaseb(bb)=table1r0(jbb(j,b),3)
            end;end;
            end;
cc=0;
       if  jbs~=0
         for c=1:length(jbs(j,:))
                  if     OPERATIONS(jbs(j,c))~=
OPERATIONS(jbs(j,ii))  & jbs(j,c)~= jbs(j,ii)  &
table1r0(jbs(j,c),3) < gtime &  (gtime +
table1r0(re(j),2)- table1r0(re(j),4)) <  (max((time(j,it-
1)+table1r0(jbs(j,c),2)), (table1r0(jbs(j,c),3)+
table1r0(jbs(j,c),2)) - table1r0(jbs(j,c),4)))
```

```
                    cc=cc+1; gjobs(cc)= jbs(j,c);
greleases(cc)=table1r0(jbs(j,c),3)
            end;end;
            end;
if  bb~=0
            mingreleaseb=min(greleaseb);
for   bbb=bb:-1:1
                    if greleaseb(bbb)==mingreleaseb
                job(j,it)=gjobb(bbb); time(j,it)=
max((time(j,it-1)+table1r0(gjobb(bbb),2)),
(table1r0(gjobb(bbb),3)+ table1r0(gjobb(bbb),2)));
   end;end;
end;
if  cc~=0
            mingreleases=min(greleases);
            for   ccc=cc:-1:1
                    if greleases(ccc)==mingreleases
job(j,it)=gjobs(ccc); time(j,it)= max((time(j,it-
1)+table1r0(gjobs(ccc),2)), (table1r0(gjobs(ccc),3)+
table1r0(gjobs(ccc),2)));
   end;end;
end;
if bb==0 & cc==0
tm(j,it) =max((time(j,it-
1)+table1r0(jbs(j,ii),2)),(time(j,rex(j))-
table1r0(re(j),2)+table1r0(jbs(j,ii),3)-
table1r0(re(j),3)+ table1r0(jbs(j,ii),2)));
            job(j,it) = jbs(j,ii); time(j,it) =tm(j,it);
end;
  else
        tm(j,it) = time(j,it-1)+table1r0(jbs(j,ii),2);
job(j,it) = jbs(j,ii); time(j,it) =tm(j,it);
  end;
end;
end;
            fprintf ('%2.0f. operation of machine
%2.0f  is jb  %2.0f\n\n', it,j, job(j,it));
      Lmaxxx(j,it) = time(j,it)-table1r0(job(j,it),4);
disp('Lmaxxx='); disp(Lmaxxx);
elseif z(j)==0 & y(j)>=1& jbb(j,1)~=0
vv=0; uu=0;  tt=0;  ttt=0; ttmin=0; jb=0; u=0;
d=0; dd=0; ddd=0; dddd=0; r=0; rr=0; rrr=0;
gtime=0;
for i=1:a
   if k(i)==j
      for b=1:length(jbb(j,:))
        if i==jbb(j,b)

vv=vv+1;tt(j,vv)=table1r0(i,3)+table1r0(i,2);
        end;
    end;
end;
end;
v(j)=vv;
for z=1:v(j)
    ttt(z)= tt(j,z);
end;
ttmin(j)=min(ttt);
for i=1:a
      if  k(i)==j
        for b=1:length(jbb(j,:))
          if  i==jbb(j,b) &  table1r0(i,3)<ttmin(j)
        uu=uu+1;jb(j,uu)=i;
      end;
   end;
end;
end;
u(j)=uu;
      for h=1:u(j)
        d(j,h)=table1r0(jb(j,h),4);
          ddd(h)= d(j,h);
      end;
      dd(j)=min(ddd);
        for g=1:u(j)
          if table1r0(jb(j,g),4)==dd(j)
            dddd=dddd+1;
            r(j,dddd)= table1r0(jb(j,g),3);
            rrr(dddd)=r(j,dddd);
          end;
        end;
        rr(j)=min(rrr);
    for g=u(j):-1:1
        if  table1r0(jb(j,g),4)==dd(j)
          if  table1r0(jb(j,g),3)==rr(j)
re(j)=0;
for aa=1:it-1
      if   OPERATIONS(jb(j,g)) ==
OPERATIONS(job(j,aa))
          re(j)=job(j,aa); rex(j)=aa;
      elseif  re(j)==0 & OPERATIONS(jb(j,g)) ~=
OPERATIONS(job(j,aa))
          re(j)=job(j,it-1);
      end;
end;
        if   OPERATIONS(jb(j,g)) ==
OPERATIONS(re(j))
gtime =
max((table1r0(jb(j,g),3)+table1r0(jb(j,g),2)),(time(j,re
x(j))-table1r0(re(j),2)+table1r0(jb(j,g),3)-
table1r0(re(j),3)+ table1r0(jb(j,g),2))) -
table1r0(jb(j,g),2);
bb=0;
     if  jbb~=0
       for b=1:length(jbb(j,:))
            if  OPERATIONS(jbb(j,b))~=
OPERATIONS(jb(j,g))  & jbb(j,b)~= jb(j,g)  &
table1r0(jbb(j,b),3) < gtime &  (gtime +
table1r0(re(j),2)- table1r0(re(j),4)) <  (max((time(j,it-
1)+table1r0(jbb(j,b),2)), (table1r0(jbb(j,b),3)+
table1r0(jbb(j,b),2)) - table1r0(jbb(j,b),4)))
            bb=bb+1; gjobb(bb)= jbb(j,b);
greleaseb(bb)=table1r0(jbb(j,b),3)
        end;end;
        end;
if bb~=0
        mingreleaseb=min(greleaseb);
 for   bbb=bb:-1:1
            if greleaseb(bbb)==mingreleaseb
          job(j,it)=gjobb(bbb); time(j,it)=
max((time(j,it-1)+table1r0(gjobb(bbb),2)),
(table1r0(gjobb(bbb),3)+ table1r0(gjobb(bbb),2)));
   end;end;
end;
if bb==0
tm(j,it)
=max((table1r0(jb(j,g),3)+table1r0(jb(j,g),2)),(time(j,
rex(j))-table1r0(re(j),2)+table1r0(jb(j,g),3)-
table1r0(re(j),3)+ table1r0(jb(j,g),2)));
        job(j,it) = jb(j,g); time(j,it) =tm(j,it);
end;
  else
    tm(j,it) = table1r0(jb(j,g),3)+table1r0(jb(j,g),2);
job(j,it) = jb(j,g); time(j,it) =tm(j,it);
 end;
    fprintf ('%2.0f. operation of machine %2.0f  is jb
%2.0f\n\n', it,j, job(j,it));
```

```
  Lmaxxx(j,it) = time(j,it)-table1r0(job(j,it),4);
disp('Lmaxxx='); disp(Lmaxxx);
 end;
                end;
            end;
   end;
zz=0;yy=0; jbs=0; due=0; jbb=0; release=0;
for i=1:a
          if k(i)==j & i~=job
             if table1r0(i,3)<=time(j,it)
                 zz=zz+1; jbs(j,zz)=i; due(j,zz)=
table1r0(jbs(j,zz),4);
              elseif table1r0(i,3)>time(j,it)
                   yy=yy+1; jbb(j,yy)=i;
release(j,yy)=table1r0(jbb(j,yy),3);
             end;
          end;
end;
z(j)=zz;y(j)=yy; disp('z(j)'); disp(z(j));  disp('y(j) ');
disp(y(j));
if zz>=1
fprintf ('%2.0f. machine jbs=\n', j);disp(jbs(j,:));
fprintf ('%2.0f. machine due=\n', j);disp(due(j,:));
else
jbs=0; due=0;
end;
if yy>=1
fprintf ('%2.0f. machine jbb=\n', j);disp(jbb(j,:));
fprintf ('%2.0f. machine release=\n',
j);disp(release(j,:));
else
jbb=0; release=0;
end;
end;
Lmax(j)= max(Lmaxxx(j,:));
if Lmax(j)>=0
   fprintf ('Lmax( %2.0f )=\n',j); disp(Lmax(j));
else
   Lmax(j)=0;  fprintf ('Lmax( %2.0f )=\n',j);
disp(Lmax(j));
end;
j=j+1;
end;
for j=1:m
if time(j,1)==0
   Lmax(j)=-1;
end;
end;
Lmaxx(itno)=max(Lmax);
for j=m:-1:1
   if Lmax(j)==Lmaxx(itno)
      Bottleneck(itno)=j;
  end;
end;
fprintf ('Bottleneck(%2.0f )=\n',itno);
disp(Bottleneck(itno));
jx=0;
for j=1:m
  if Bottleneck(itno)== j
        mchh= mch(j);
          for i=1:mchh
              jx(i)=job(j,i);
          end;
   end;
end;
for i=1:mchh-1
  if  sc2(jx(i))==0
     sc2(jx(i))= jx(i+1);
  end;
end;
for i=2:mchh
    if  pr2(jx(i))==0
      pr2(jx(i))= jx(i-1);
    end;
end;
for i=1:a
    tes(i)=0;
    tef(i)=0;
end;
for i=1:a
   if pr1(i)<i & pr2(i)<i
   if pr1(i)==0&pr2(i)==0
     tes(i)=0;
     tef(i)=t(i);
   elseif pr1(i)==0 & pr2(i)~=0 & tef(pr2(i))~=0
      tes1(i)=0; tes2(i)=tef(pr2(i));
     tef1(i)= t(i); tef2(i)=t(i)+tef(pr2(i));
      tef(i)=max(tef1(i),tef2(i));
      tes(i)=max(tes1(i),tes2(i));
   elseif pr2(i)==0 & tef(pr1(i))~=0 & pr1(i)~=0
     tef(i)=t(i)+tef(pr1(i));
     tes(i)=tef(pr1(i));
   elseif pr1(i) ~=0 & pr2(i)~=0 & tef(pr1(i))~=0 &
tef(pr2(i))~=0
      tef1(i)=t(i)+tef(pr1(i));tes1(i)=tef(pr1(i));
      tef2(i)=t(i)+tef(pr2(i));tes2(i)=tef(pr2(i));
      tef(i)=max(tef1(i),tef2(i));
      tes(i)=max(tes1(i),tes2(i));
   end;
   end;
end;
for i=1:a
     if tef(i)==0 | pr1(i)>i | pr2(i)>i
   if pr1(i)==0&pr2(i)==0
     tes(i)=0;
     tef(i)= t(i);
   elseif pr1(i)==0 & pr2(i)~=0 & tef(pr2(i))~=0
      tes1(i)=0; tes2(i)=tef(pr2(i));
     tef1(i)= t(i); tef2(i)=t(i)+tef(pr2(i));
      tef(i)=max(tef1(i),tef2(i));
      tes(i)=max(tes1(i),tes2(i));
   elseif pr2(i)==0& tef(pr1(i)) ~=0& pr1(i)~=0
     tef(i)=t(i)+tef(pr1(i));
     tes(i)=tef(pr1(i));
   elseif pr1(i) ~=0 & pr2(i)~=0 & tef(pr1(i)) ~=0 &
tef(pr2(i)) ~=0
      tef1(i)=t(i)+tef(pr1(i));tes1(i)=tef(pr1(i));
      tef2(i)=t(i)+tef(pr2(i));tes2(i)=tef(pr2(i));
      tef(i)=max(tef1(i),tef2(i));
      tes(i)=max(tes1(i),tes2(i));
    end;
   end;
end;
fi=0;
while fi==0
for i=1:a
   if tef(i)==0
    if pr1(i)==0&pr2(i)==0
     tes(i)=0;
     tef(i)= t(i);
   elseif pr1(i)==0 & pr2(i)~=0 & tef(pr2(i))~=0
      tes1(i)=0; tes2(i)=tef(pr2(i));
     tef1(i)= t(i); tef2(i)=t(i)+tef(pr2(i));
      tef(i)=max(tef1(i),tef2(i));
      tes(i)=max(tes1(i),tes2(i));
   elseif pr2(i)==0 & tef(pr1(i)) ~=0& pr1(i)~=0
```

APPENDIX B

The Computational Results Of The Test Problems

Table B.1 The results of the 3x5x[6,30] test problems

	MIP		MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	294	856	327	13	332	13	309	17	348	26	334	22	347	26	316	24	382	25	366	24
2	333	108	349	12	354	12	354	16	369	28	415	22	412	24	417	24	371	24	407	24
3	292	282	317	13	326	11	319	16	327	27	391	23	341	24	378	23	327	26	378	23
4	333	81	352	12	352	12	352	13	366	28	355	25	370	28	359	27	382	23	355	24
5	311	159	341	13	357	11	350	16	371	26	375	23	331	28	362	25	371	29	380	24
6	297	72	328	12	328	11	328	15	361	25	387	24	395	26	387	25	402	27	392	24
7	277	90	291	12	284	11	284	15	289	24	289	25	289	24	289	26	289	24	289	23
8	302	81	323	12	317	11	323	14	323	25	322	24	315	25	343	24	323	26	332	23
9	294	306	331	13	322	12	320	16	370	27	379	25	375	26	352	27	341	29	356	26
10	139	90	153	11	153	11	153	14	174	24	188	24	179	24	181	23	176	20	172	22
11	124	83	127	10	131	10	131	11	131	25	152	22	127	23	137	25	167	25	162	23
12	370	1297	393	13	397	13	416	17	381	28	411	25	414	25	416	26	426	26	445	26
13	305	66	321	12	321	11	321	14	310	24	337	23	345	25	340	26	310	25	343	23
14	312	902	336	14	327	12	318	17	320	26	318	25	329	28	331	27	321	28	336	25
15	265	52	267	11	267	11	271	13	267	23	267	24	265	23	265	23	265	23	281	23
16	211	91	216	11	216	11	216	12	249	25	239	23	225	25	237	23	225	23	224	22
17	196	81	210	11	201	11	210	12	206	23	229	24	219	25	224	23	219	23	248	23
18	324	122	339	13	349	12	334	15	384	27	383	24	383	25	367	24	384	25	395	24
19	338	301	362	14	348	12	362	16	348	26	401	24	363	26	450	24	371	25	393	24
20	228	58	242	12	249	11	242	13	228	27	250	23	228	27	228	25	242	25	258	23

Table B.2 The results of the 3x5x[6,40] test problems

	MIP		MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	439	797	451	12	475	13	468	21	464	28	489	26	460	26	454	26	468	28	459	26
2	323	449	339	11	339	12	339	15	371	27	381	25	330	24	331	24	371	28	384	26
3	417	1046	458	13	467	13	466	19	467	28	447	27	444	27	467	24	458	26	492	26
4	375	143	395	12	411	12	411	17	412	29	413	27	445	29	463	24	412	24	402	23
5	414	707	421	13	417	15	418	20	477	27	526	25	478	29	537	26	427	26	418	25
6	318	343	373	12	357	12	357	16	415	27	416	26	412	24	452	23	375	24	399	25
7	357	1177	382	13	374	14	379	19	407	29	403	27	398	27	431	25	394	25	395	26
8	394	129	419	12	419	12	416	16	412	25	445	23	434	26	434	24	419	24	403	22
9	205	973	208	12	208	12	208	15	210	26	238	24	210	26	251	24	218	24	249	23
10	276	126	306	11	301	12	301	15	301	24	289	25	324	24	283	24	301	24	318	23
11	275	99	297	11	283	12	302	17	293	25	291	26	291	24	282	24	293	26	317	26
12	268	700	278	12	277	12	275	15	298	25	304	25	275	24	282	24	293	26	287	26
13	324	109	343	13	343	12	343	16	399	26	449	26	365	24	409	25	396	26	379	27
14	417	9808	465	16	476	15	471	21	486	29	496	27	498	26	491	26	506	28	461	28
15	367	624	396	12	396	12	396	17	392	25	408	25	404	25	436	26	403	26	403	27
16	250	120	267	11	267	11	263	14	263	25	276	25	279	23	271	24	280	25	276	25
17	389	15825	423	15	422	15	412	20	495	29	468	27	465	26	470	26	509	29	427	28
18	227	141	229	11	229	12	229	14	245	26	272	27	243	23	287	23	229	25	241	25
19	432	245	471	13	496	13	477	19	489	27	495	27	507	25	506	25	489	26	507	27
20	269	229	286	11	289	12	286	14	295	24	326	25	326	23	294	24	289	24	312	23

Table B.3 The results of the 3x10x[11,20] test problems

	MIP		MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	250	23	259	13	259	13	259	22	261	28	267	27	262	26	262	26	261	26	267	25
2	188	31	201	13	196	11	188	20	188	25	199	25	200	25	188	27	203	25	200	25
3	251	20	251	13	251	12	251	22	277	28	251	27	251	29	303	26	251	26	257	25
4	199	30	216	12	211	11	211	19	225	28	256	25	205	28	244	27	206	25	221	26
5	239	26	239	13	241	12	241	18	241	28	239	27	239	29	251	27	241	27	262	26
6	168	18	168	12	168	11	168	18	168	26	168	27	168	25	168	26	168	27	168	25
7	164	19	169	12	169	11	172	17	172	28	189	27	172	29	200	27	172	27	172	26
8	193	13	193	12	193	10	193	17	193	26	193	26	193	24	193	25	193	26	193	23
9	198	27	198	13	203	13	203	18	203	25	205	25	214	25	217	24	198	25	204	24
10	168	13	170	12	170	10	178	17	178	24	180	25	194	26	216	24	180	26	171	23
11	202	25	202	13	202	12	202	18	202	27	243	26	242	28	229	25	202	26	202	26
12	240	28	248	13	242	12	248	20	265	27	262	27	252	29	262	25	272	27	254	25
13	192	24	194	12	192	11	192	22	201	25	249	26	194	25	256	24	201	25	223	25
14	226	41	235	13	229	12	229	20	229	25	229	24	229	26	303	25	229	27	247	26
15	211	27	217	13	217	12	217	20	217	26	230	24	217	28	282	25	217	27	216	26
16	174	16	174	12	174	10	174	17	177	24	177	23	236	26	229	24	177	26	177	24
17	175	13	183	12	183	11	183	18	228	26	201	24	183	27	228	24	183	26	201	25
18	162	18	162	12	162	10	162	17	162	26	162	24	180	24	180	24	203	26	172	24
19	160	16	160	12	160	11	160	18	160	25	162	22	160	25	160	24	160	26	160	24
20	185	16	185	12	186	11	186	19	186	26	186	25	187	26	186	25	186	26	186	25

Table B.4 The results of the 3x10x[11,30] test problems

	MIP		MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	329	60	340	16	340	14	340	30	355	31	360	30	338	39	336	28	396	28	340	26
2	279	59	287	15	287	12	287	26	287	29	287	28	287	39	291	27	291	27	287	25
3	333	105	347	17	336	14	336	37	348	33	355	30	348	34	446	30	348	31	346	29
4	241	59	241	16	241	11	241	23	292	28	315	27	308	27	271	27	253	26	244	25
5	234	77	239	15	234	11	235	25	242	27	260	24	235	28	260	25	242	27	260	24
6	280	50	286	15	286	11	286	20	286	27	286	30	286	27	286	29	286	32	297	34
7	227	54	231	16	231	13	227	30	277	28	231	28	247	26	240	27	227	32	241	34
8	266	50	278	16	278	14	271	32	296	29	311	29	328	26	285	26	298	32	298	30
9	229	65	240	14	229	11	229	20	229	26	236	27	229	28	258	25	229	29	241	26
10	296	66	303	16	313	13	309	30	345	29	384	30	350	29	427	29	310	33	340	31
11	313	58	324	15	313	11	313	37	320	28	377	30	330	28	431	29	313	30	313	31
12	228	44	233	14	249	11	233	22	239	26	262	27	251	27	267	26	235	26	235	26
13	182	56	185	13	185	12	182	21	185	25	205	25	235	24	240	24	185	27	185	25
14	280	50	280	13	280	11	280	21	297	25	301	24	281	25	280	24	281	27	280	23
15	368	54	385	15	377	12	385	28	377	26	377	25	374	27	388	26	390	30	373	26
16	301	46	301	15	302	12	301	25	364	27	342	26	314	27	372	27	339	30	372	26
17	348	46	363	14	348	12	348	28	348	26	379	24	357	29	369	25	348	27	379	25
18	286	45	301	15	286	11	299	27	289	29	304	27	311	29	307	27	286	29	320	26
19	290	46	291	15	291	11	291	24	291	26	291	26	290	26	299	28	291	34	291	29
20	296	47	303	15	303	11	302	30	332	35	323	30	302	27	346	26	317	32	321	30

Table B.5 The results of the 5x3x[4,6] test problems

	MIP		MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_BMCH (2 window)		MRHP_BJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (3 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	122	731	132	12	133	9	146	11	139	24	138	22	143	29	145	22	144	25	145	26
2	133	1520	133	11	141	9	141	11	133	23	139	24	148	24	139	25	147	22	163	22
3	94	67	109	11	109	10	107	12	110	23	122	25	116	24	102	21	114	22	121	23
4	116	87	128	11	128	10	128	11	137	25	135	25	138	26	136	21	133	22	133	23
5	135	1426	143	11	143	9	143	11	148	24	143	25	146	23	148	22	139	22	181	24
6	141	638	143	11	143	9	143	11	155	23	143	25	146	24	169	22	149	22	181	24
7	118	151	118	11	127	10	118	11	123	24	123	25	133	24	134	22	131	22	172	24
8	93	47	99	12	99	9	95	11	98	24	115	22	99	28	99	25	113	25	98	22
9	100	523	100	10	100	9	100	11	103	24	112	24	123	24	100	22	108	22	100	24
10	118	51	119	11	119	10	119	11	129	24	133	24	139	25	143	22	136	22	123	24
11	88	21	100	11	99	10	97	11	112	24	124	24	103	22	122	22	121	22	104	22
12	118	419	118	11	118	9	128	11	137	24	123	23	148	25	137	22	137	22	146	23
13	94	20	99	11	99	9	107	11	94	22	107	22	98	22	103	22	105	22	110	22
14	82	24	89	10	91	9	91	11	89	22	89	22	89	24	100	22	100	22	111	22
15	104	330	104	11	104	9	104	11	104	22	104	23	107	25	104	22	104	22	112	22
16	117	572	117	11	120	10	120	11	133	21	133	23	146	27	130	22	121	22	134	22
17	109	99	113	11	113	9	113	11	127	22	127	23	124	25	109	22	110	22	128	22
18	129	861	142	10	140	9	130	11	153	22	153	23	153	24	155	22	155	22	156	22
19	113	39	119	10	119	9	113	11	129	24	129	22	127	24	127	22	127	22	138	22
20	105	38	110	12	110	9	110	11	115	25	120	22	121	29	120	24	129	23	133	22

Table B.6 The results of the 5x3x[4,9] test problems

	MIP		MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_BMCH (2 window)		MRHP_BJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (3 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	145	837	151	11	151	10	162	11	180	23	186	24	186	20	203	23	175	23	175	22
2	124	3708	134	11	134	10	134	11	143	22	141	25	146	20	151	23	142	22	135	21
3	148	22023	148	11	148	11	148	11	161	22	148	25	166	21	157	22	155	23	162	22
4	142	3731	150	11	150	10	150	11	163	23	157	22	168	22	186	22	172	23	182	22
5	113	1289	125	12	113	10	113	11	142	24	138	24	144	23	134	24	122	23	149	22
6	137	5184	139	11	139	10	137	12	151	23	137	22	144	22	152	22	167	23	144	22
7	104	325	117	11	106	11	121	11	117	27	148	27	117	27	117	25	159	25	117	24
8	150	6439	154	11	154	11	154	11	157	23	197	23	154	22	171	22	177	22	208	22
9	121	119	134	11	137	10	137	11	179	22	135	22	158	22	147	22	153	22	167	22
10	146	4194	170	11	170	11	170	11	170	22	170	23	170	22	174	22	171	24	170	23
11	172	25089	180	11	180	11	180	11	191	22	180	22	222	22	218	22	190	24	217	23
12	128	4380	131	12	131	11	137	11	157	24	157	27	157	24	157	25	157	25	183	23
13	147	4967	150	11	150	10	150	11	161	23	161	22	161	22	165	22	161	22	161	22
14	152	9596	179	11	179	10	179	11	199	22	199	22	184	22	201	22	201	22	191	23
15	138	929	150	11	150	10	150	11	148	22	146	22	176	22	182	22	166	22	142	22
16	212	300	239	11	239	11	230	12	216	23	219	22	243	22	222	23	233	23	246	23
17	151	12693	151	11	151	10	151	12	164	22	171	22	154	22	178	22	179	23	174	22
18	164	2296	181	11	185	11	185	11	197	27	197	27	197	27	199	24	220	24	199	24
19	138	4396	153	11	144	10	153	11	160	23	152	22	158	22	163	22	156	22	153	23
20	149	24253	157	11	156	10	156	12	166	23	156	22	167	22	160	22	163	24	156	22

Table B.7 The results of the 5x5x[6,8] test problems

	MIP		MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_BMCH (2 window)		MRHP_BJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (3 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	120	56	129	11	149	11	140	12	140	23	161	23	143	24	146	22	165	22	139	24
2	96	14	108	12	108	11	107	12	104	22	104	22	111	23	112	22	133	21	121	23
3	117	182	122	11	117	11	117	12	137	22	144	22	127	23	128	22	135	22	141	23
4	118	112	125	12	121	10	126	11	133	23	140	22	133	23	143	22	144	22	148	23
5	99	75	109	13	105	10	113	11	109	22	108	23	106	23	122	21	108	23	129	22
6	124	122	133	12	134	10	135	11	137	23	142	23	158	23	140	24	124	23	148	23
7	127	51	128	11	134	10	134	11	148	22	159	22	148	23	159	23	145	22	145	22
8	100	37	102	11	102	10	102	11	108	22	122	22	121	23	127	22	113	22	137	23
9	135	34	152	11	155	10	154	12	148	24	178	23	161	23	155	22	143	23	175	22
10	104	39	113	11	112	10	122	11	109	24	109	22	116	22	125	23	122	23	116	23
11	87	39	88	11	88	10	88	11	97	23	104	22	95	22	93	23	93	23	99	22
12	122	76	135	11	135	10	130	11	126	23	133	22	164	22	129	23	145	23	140	23
13	120	66	128	13	128	10	128	11	130	23	130	23	176	24	130	24	131	23	147	23
14	115	76	125	11	125	10	125	11	120	22	124	23	127	22	120	23	120	23	124	24
15	109	57	117	12	117	11	117	11	117	22	117	26	125	23	126	23	120	24	132	23
16	96	66	101	11	101	10	101	11	127	22	132	23	127	22	135	22	121	23	119	22
17	130	65	142	11	139	10	142	11	142	23	153	23	168	23	142	22	148	23	190	23
18	109	172	109	11	109	10	109	11	117	22	118	22	116	22	109	23	118	23	123	22
19	128	73	130	11	142	10	140	11	138	23	149	23	138	23	133	23	162	23	174	22
20	105	44	108	11	117	10	117	11	112	22	112	23	119	22	113	21	121	23	125	22

Table B.8 The results of the 5x5x[6,10] test problems

	MIP		MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_BMCH (2 window)		MRHP_BJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (3 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	163	246	175	12	171	11	171	13	181	23	175	23	192	23	183	21	172	24	214	21
2	131	458	144	13	148	11	147	13	155	24	163	22	159	23	155	23	143	23	165	19
3	147	465	150	12	150	10	150	12	177	23	168	23	165	23	175	22	171	23	163	21
4	139	92	142	12	142	11	159	11	165	24	179	23	142	23	189	21	158	23	186	21
5	151	243	164	12	164	11	164	12	167	24	166	23	182	23	200	22	189	24	207	23
6	135	64	166	11	166	11	166	11	166	23	176	23	174	23	159	24	162	23	149	21
7	101	66	105	11	116	11	116	11	124	23	118	22	127	22	139	24	125	23	145	21
8	183	8617	188	12	188	10	191	11	208	24	190	23	206	23	212	23	184	23	203	22
9	123	143	128	11	130	11	127	12	144	23	159	23	147	23	151	26	154	23	151	21
10	145	185	171	13	171	11	172	11	168	25	191	26	205	29	174	23	209	23	170	25
11	145	108	152	12	152	11	153	11	152	23	150	26	170	29	158	23	172	24	158	25
12	108	69	114	11	112	11	112	11	125	23	111	23	111	23	114	22	149	23	111	21
13	128	125	152	11	151	11	137	11	146	23	140	23	145	23	153	22	147	23	155	22
14	158	421	188	12	171	10	178	11	180	23	171	23	192	23	191	23	178	23	199	21
15	137	83	151	12	151	11	160	11	179	23	168	23	180	23	166	22	174	23	181	21
16	136	235	146	11	149	11	146	11	149	23	149	23	182	23	172	21	162	23	163	21
17	129	107	152	11	145	10	160	11	176	23	144	23	185	23	160	22	179	23	150	21
18	131	82	144	11	142	10	144	11	143	23	148	23	172	23	159	22	150	23	171	22
19	149	23	159	11	164	10	155	12	159	23	159	23	173	22	178	22	169	23	188	21
20	108	53	116	11	121	10	121	11	126	23	118	22	121	22	128	22	130	22	125	21

Table B.9 The results of 10x5x[6,10] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	217	12	219	11	211	15	233	23	205	24	220	21	228	24	229	24	233	24
2	235	12	221	11	221	15	239	23	246	23	236	21	249	24	245	23	272	23
3	237	12	237	12	237	15	246	23	256	23	253	22	252	24	252	23	264	23
4	222	12	238	12	233	15	241	23	241	24	216	23	241	25	224	23	246	24
5	241	13	241	12	265	17	269	28	260	24	282	23	257	26	272	24	284	25
6	177	12	177	11	186	14	187	23	210	23	198	22	232	23	217	23	219	23
7	204	12	214	12	206	15	219	23	217	24	202	23	225	24	225	23	239	24
8	240	13	240	12	238	16	258	23	243	24	272	23	259	24	252	24	261	25
9	235	12	235	12	232	16	252	23	267	24	239	23	239	24	238	24	277	24
10	228	12	236	12	235	16	239	23	261	24	262	23	297	24	248	23	249	24

Table B.10 The results of 10x10x[11,15] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	267	23	266	23	262	49	273	36	324	39	294	35	337	41	312	37	324	45
2	240	23	276	21	258	44	228	35	257	38	266	36	259	40	273	36	268	39
3	262	24	258	22	259	44	285	39	316	41	292	37	336	40	281	39	324	44
4	227	24	249	22	246	42	260	36	266	39	235	36	261	40	269	39	267	45
5	263	25	260	24	261	46	298	36	275	44	262	38	305	44	289	38	316	44
6	247	25	236	21	222	40	253	35	284	44	234	36	266	39	241	35	288	38
7	291	24	280	25	267	45	313	34	309	53	268	36	316	40	305	36	341	41
8	296	25	309	24	281	49	310	37	318	46	295	44	340	42	297	38	322	44
9	280	25	272	21	270	44	289	37	295	44	291	36	294	39	292	36	309	40
10	243	23	261	21	233	41	271	35	290	41	269	37	285	38	248	36	290	42

Table B.11 The results of 15x10x[11,15] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	413	38	391	35	378	82	434	54	413	59	422	53	413	59	446	56	421	58
2	339	36	324	33	334	74	354	52	359	57	339	59	355	58	342	50	367	68
3	304	33	321	31	308	68	349	48	368	52	334	65	364	54	336	49	350	68
4	336	38	329	32	341	77	369	50	392	56	340	59	355	57	388	54	376	62
5	332	35	332	31	315	69	363	48	388	54	343	49	382	53	376	51	387	62
6	322	37	326	32	321	74	344	52	347	57	336	57	340	56	333	54	397	61
7	331	36	325	32	322	71	304	48	342	52	336	50	329	52	328	51	330	61
8	328	35	334	31	337	71	329	48	357	53	337	54	349	53	356	48	384	57
9	368	41	344	35	349	85	389	52	416	56	366	55	350	56	367	66	376	64
10	388	39	404	34	379	84	408	52	424	59	390	50	398	56	428	56	435	63

Table B.12 The results of 15x15x[16,20] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	367	114	363	94	384	231	382	123	419	135	467	125	419	141	386	126	429	153
2	370	99	372	83	333	203	380	111	368	132	424	125	396	117	381	118	408	137
3	366	104	364	85	349	223	420	112	421	132	383	129	388	140	411	124	438	150
4	403	99	364	85	376	221	400	110	402	126	420	120	408	123	415	122	434	131
5	381	105	355	92	374	220	374	112	372	133	351	122	398	138	369	121	366	145
6	374	100	355	91	340	213	368	112	384	125	414	124	439	121	384	136	426	136
7	357	98	368	86	373	242	414	111	401	139	418	121	435	134	447	133	433	147
8	366	99	379	89	400	240	384	116	397	136	399	127	425	125	412	129	457	146
9	350	96	395	89	392	235	407	115	410	135	391	119	427	137	368	124	370	136
10	371	97	337	87	367	228	368	108	419	133	400	119	414	133	381	122	389	132

Table B.13 The results of 20x15x[16,20] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	450	176	415	143	436	372	507	181	501	214	497	192	487	218	495	203	515	267
2	461	171	481	152	442	369	477	171	483	204	468	183	544	211	503	187	540	234
3	460	176	434	126	463	364	443	160	468	183	505	169	488	194	468	181	492	211
4	405	162	466	124	426	330	427	162	441	201	410	165	443	203	401	167	471	219
5	418	183	448	145	427	374	481	187	457	209	468	181	455	224	451	195	488	234
6	429	160	466	126	447	359	468	161	447	203	433	164	471	205	451	178	460	200
7	425	163	419	132	403	336	468	163	483	198	444	168	450	209	457	174	475	192
8	431	181	444	144	444	384	457	179	475	216	486	178	499	220	531	194	561	234
9	487	180	452	144	450	361	449	181	485	204	462	185	474	202	499	187	521	223
10	429	162	424	131	401	301	463	166	451	194	475	170	471	197	467	164	518	200

Table B.14 The results of 20x20x[21,25] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	470	375	487	307	479	833	468	373	507	464	493	432	536	476	526	398	522	471
2	484	373	484	300	484	821	506	375	549	459	485	432	590	446	485	386	551	479
3	505	371	505	313	481	816	493	371	492	449	560	415	542	462	509	417	569	487
4	483	359	454	284	442	773	477	363	550	430	504	382	529	433	508	377	567	494
5	479	386	468	300	466	799	452	366	481	439	481	379	499	447	477	377	537	463
6	506	371	466	301	503	844	505	396	525	496	475	400	506	471	554	424	584	488
7	473	349	480	289	471	805	513	377	556	473	514	395	568	472	519	424	578	477
8	509	358	487	289	466	769	457	353	522	469	488	400	554	481	515	376	535	466
9	482	368	532	309	471	821	503	382	484	461	489	390	532	465	541	434	534	521
10	473	330	440	276	445	741	467	377	464	440	492	354	524	427	511	377	561	450

Table B.15 The results of 30x15x[16,20] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	552	322	544	257	577	681	554	323	531	373	503	332	576	387	559	355	546	418
2	638	331	638	262	599	730	611	325	663	400	619	359	646	415	657	339	619	427
3	546	326	510	245	531	702	546	305	568	384	571	326	600	400	610	380	572	402
4	606	315	584	257	558	700	570	317	583	396	616	322	650	413	633	348	630	411
5	549	313	537	254	519	652	561	321	642	429	637	321	607	403	574	326	608	400
6	589	339	577	267	590	757	619	330	627	407	610	376	635	431	589	345	651	421
7	580	362	572	265	584	761	619	382	615	371	615	417	627	386	616	354	673	432
8	537	315	555	245	530	669	536	306	561	362	597	307	599	363	600	328	592	417
9	528	321	538	258	554	677	601	316	605	386	574	316	590	397	574	376	589	412
10	596	315	531	247	553	676	542	311	559	382	537	300	544	382	576	330	585	416

Table B.16 The results of 30x20x[21,25] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	607	753	626	588	603	1541	585	720	622	888	636	740	609	951	651	781	647	974
2	640	775	601	633	621	1639	634	722	682	878	643	798	640	939	650	766	664	945
3	575	747	609	585	581	1546	570	715	610	836	601	786	609	913	614	747	626	956
4	612	771	626	653	667	1647	675	727	705	894	645	802	656	931	674	802	673	959
5	623	793	639	619	642	1684	702	757	711	900	697	970	728	974	667	819	745	1099
6	645	770	623	621	650	1602	643	741	635	846	645	792	612	925	663	767	669	954
7	571	753	628	594	617	1565	640	711	652	857	621	739	653	924	626	819	692	995
8	648	757	623	602	603	1561	626	716	851	1390	602	870	647	864	649	770	666	931
9	651	808	614	634	617	1644	679	759	696	895	645	845	690	968	665	834	723	972
10	624	763	571	601	629	1564	584	719	605	882	609	739	628	943	613	779	645	931

Table B.17 The results of 50x15x[16,20] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	789	840	845	664	800	1667	889	804	938	963	839	771	864	905	873	857	945	1097
2	806	855	831	663	837	1730	822	809	849	981	870	784	878	920	918	857	903	1025
3	839	830	821	620	837	1695	871	770	906	906	843	773	883	913	829	813	929	1047
4	960	887	967	686	1085	2425	957	893	1042	968	954	809	949	981	951	847	978	1064
5	770	821	772	645	776	1565	806	770	831	889	819	735	774	868	832	838	843	1082
6	857	883	855	692	878	1839	865	821	886	927	848	789	914	967	898	872	899	1049
7	847	884	863	667	841	1748	890	1016	991	952	935	831	908	1009	880	884	965	1069
8	819	836	785	639	781	1675	810	771	879	883	832	771	831	886	818	865	882	1067
9	801	774	817	618	802	1635	826	719	865	861	815	699	846	850	839	799	834	943
10	785	796	764	615	787	1626	771	737	818	843	795	775	796	883	794	771	833	968

Table B.18 The results of 50x20x[21,25] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	942	2011	891	1509	835	3895	868	1728	894	2098	836	1714	849	2090	868	1856	904	2280
2	894	2006	864	1548	859	3997	922	1792	931	2084	915	1900	934	2335	913	1932	965	2335
3	888	1984	887	1491	845	3906	963	1841	1008	2100	903	1768	1006	2044	940	1900	981	2419
4	892	2004	906	1577	864	3837	845	1834	826	2084	858	1949	862	2077	855	1939	898	2355
5	795	1978	796	1553	775	3724	807	1760	829	2071	828	1792	858	2119	856	1949	873	2414
6	821	1861	769	1412	796	3577	807	1662	834	1989	818	1647	890	2186	855	1887	888	2232
7	896	1938	910	1466	888	3889	852	1701	876	2074	850	1748	890	2122	887	1856	894	2205
8	899	1904	829	1456	812	3680	832	1763	855	2090	812	1762	825	1986	833	1863	885	2386
9	879	2025	874	1549	909	3914	913	1825	989	2176	910	1909	944	2166	959	2082	988	2530
10	750	2000	848	1590	840	3981	871	1787	977	2137	863	1763	883	2277	840	1911	862	2293

Table B.19 The results of 100x15x[16,20] test problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	1474	3620	1473	2548	1482	6274	1582	2892	1669	3255	1550	2872	1491	3186	1650	3105	1720	3645
2	1470	3401	1469	2448	1471	6323	1503	2784	1537	3133	1504	3001	1532	3196	1515	3028	1656	3734
3	1580	3424	1574	2586	1574	6586	1574	2832	1628	3212	1603	2880	1591	3214	1592	3083	1682	3772
4	1535	3345	1486	2498	1488	6344	1653	2831	1705	3263	1583	2982	1645	3297	1649	3129	1572	3790
5	1390	3189	1374	2345	1373	5827	1413	2763	1458	3186	1398	2573	1437	3020	1434	2898	1503	3554
6	1567	3354	1564	2489	1550	5973	1543	2835	1539	3141	1550	2755	1542	3289	1627	3190	1561	3804
7	1507	3364	1505	2537	1490	6088	1603	2917	1638	3440	1550	2976	1554	3254	1596	3140	1731	3931
8	1568	3479	1557	2587	1557	6208	1678	2872	1707	3362	1601	2880	1645	3380	1589	3166	1723	3813
9	1472	3161	1424	2355	1429	5652	1600	2743	1636	3125	1597	2640	1528	2947	1574	3022	1614	3653
10	1508	3262	1502	2492	1502	5980	1649	2834	1643	3278	1537	2731	1543	3306	1629	3111	1649	3894

Table B.20 The results of 100x20x[21,25] test problems for large size problems

	MSBH		MSBH_TML		MSBH_TOTINF		MRHP_FJOB (2 window)		MRHP_FJOB (3 window)		MRHP_BMCH (2 window)		MRHP_BMCH (3 window)		MRHP_BJOB (2 window)		MRHP_BJOB (3 window)	
prb.	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**	**Cmax**	**time**
1	1488	8371	1499	5990	1489	14578	1531	6682	1529	7234	1538	6888	1498	7549	1562	7092	1573	8292
2	1456	8081	1408	5788	1409	14034	1541	6576	1585	7197	1539	6356	1563	7391	1521	7034	1507	8365
3	1441	8197	1430	6066	1425	14145	1500	6648	1530	7381	1499	6644	1488	7325	1515	7153	1578	8416
4	1435	8114	1371	5658	1375	13453	1500	6441	1527	7076	1429	6350	1507	7155	1475	6676	1506	7916
5	1534	8224	1481	6050	1474	14502	1570	6854	1578	7281	1470	6529	1613	7639	1500	6900	1574	8319
6	1513	8146	1487	5799	1481	13917	1649	6612	1602	7465	1602	6571	1699	7473	1573	7088	1732	8241
7	1574	8418	1487	6088	1497	14611	1487	6732	1637	7676	1523	6945	1564	7976	1554	7775	1519	8000
8	1586	7806	1471	5667	1468	13600	1543	6445	1578	7194	1589	6369	1586	7025	1513	7183	1537	8532
9	1557	8509	1538	6137	1536	14728	1652	6792	1687	7488	1598	6989	1600	7000	1693	7404	1824	8707
10	1456	8015	1443	5690	1453	13656	1512	6798	1540	6803	1462	6247	1495	7284	1546	6960	1600	7034

Table B.21 The results of 10x10x[11,15] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BJOB 2 window		MRHP_BJOB 3 window	
1	1198	35	1243	43	1453	37	1465	45	1406	42	1382	39	1516	38	1499	40
2	1184	37	1311	37	1534	34	1561	37	1440	34	1379	36	1529	35	1515	37
3	1336	37	1313	37	1370	33	1514	36	1355	38	1466	37	1380	35	1514	35
4	1232	37	1335	40	1394	36	1478	37	1326	34	1380	42	1418	36	1526	39
5	1239	45	1380	35	1458	33	1576	37	1414	34	1359	35	1414	34	1481	44
6	1313	35	1446	37	1736	34	1590	37	1228	35	1253	37	1485	34	1466	41
7	1135	34	1127	34	1414	34	1525	38	1390	36	1316	37	1383	34	1369	35
8	1559	34	1660	38	1613	35	1692	38	1646	37	1583	41	1575	36	1643	38
9	1133	34	1417	38	1361	33	1505	35	1274	36	1421	50	1297	35	1437	36
10	1158	36	1206	37	1469	35	1601	38	1418	35	1512	42	1534	36	1512	33

Table B.22 The results of 10x15x[16,20] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BJOB 2 window		MRHP_BJOB 3 window	
1	1506	73	1728	76	1590	73	1624	82	1521	77	1741	81	1496	70	1496	71
2	1420	65	1686	78	1586	71	1609	80	1501	79	1568	84	1690	72	1694	76
3	1432	72	1708	77	1580	67	1729	82	1417	72	1490	83	1555	72	1742	91
4	1452	70	1650	77	1564	69	1729	74	1631	95	1618	78	1618	65	1672	82
5	1318	66	1462	79	1495	64	1580	78	1467	79	1530	88	1521	67	1589	73
6	1457	80	1712	75	1598	72	1869	77	1574	76	1679	86	1596	75	1813	87
7	1491	82	1692	83	1582	69	1711	75	1597	78	1497	90	1567	66	1743	74
8	1456	71	1685	74	1699	64	1798	74	1641	71	1832	81	1756	68	1798	74
9	1479	70	1718	76	1557	65	1656	76	1522	75	1673	86	1573	67	1706	80
10	1316	71	1466	74	1564	66	1546	79	1339	69	1528	80	1528	68	1554	73

Table B.23 The results of 10x20x[21,25] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BJOB 2 window		MRHP_BJOB 3 window	
1	1772	139	1973	147	1831	142	1740	177	1828	161	2000	157	1869	149	1869	157
2	1785	138	1936	148	1722	162	1766	193	1662	155	1828	172	1710	159	1764	164
3	1686	140	1928	150	1771	145	1803	162	1614	154	1767	150	1735	155	1757	139
4	1662	140	1898	147	1892	153	1796	170	1613	146	1927	163	1702	156	1777	186
5	1558	141	1920	150	1543	148	1631	210	1499	165	1839	167	1558	156	1663	166
6	1662	152	1938	150	1805	160	1892	194	1737	141	2040	173	1777	173	1855	150
7	1616	143	1947	139	1655	137	1700	162	1509	160	1746	170	1626	149	1683	141
8	1593	141	1882	138	1728	151	1591	159	1736	156	1578	167	1556	163	1591	163
9	1664	142	1891	176	1653	147	1806	167	1568	159	1882	191	1697	162	1758	159
10	1633	137	2060	152	1688	172	1679	176	1631	153	1789	168	1692	170	1676	179

Table B.24 The results of the MRHP_SR for the 20x10x[11,15] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_SR 5 window	
1	2176	73	2217	91	2281	111
2	2041	68	1952	77	2175	96
3	1921	64	2096	72	2241	98
4	2044	66	2036	77	2376	89
5	2225	65	2058	80	2164	103
6	1948	64	2031	77	2223	98
7	2046	68	1881	77	2243	100
8	2060	69	2045	73	2321	99
9	2050	67	2143	76	2296	99
10	1941	67	1851	78	2104	97

Table B.25 The results of the MRHP methods for the 20x10x[11,15] problems for special reentrant job shop

prb	MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_FJOB 5 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BMCH 5window		MRHP_BJOB 2 window		MRHP_BJOB 3 window		MRHP_BJOB 5 window	
1	3038	74	2991	85	3291	111	2737	81	2608	87	2629	102	2851	78	3269	98	3393	106
2	2745	65	2739	76	2971	99	2527	73	2537	80	2645	94	2656	82	2584	79	2855	92
3	2705	67	2858	78	2962	94	2531	81	2455	84	2575	103	2734	71	2693	73	2966	91
4	2786	66	2900	74	2986	100	2747	83	2576	84	2694	96	2726	80	2763	69	3036	89
5	2984	67	2975	78	3103	102	2645	78	2672	81	2708	93	2769	69	2990	74	3020	99
6	2666	67	2805	79	3058	103	2504	76	2608	80	2639	104	2702	75	2673	84	3100	100
7	2646	70	2790	80	2866	102	2352	76	2586	84	2520	101	2652	74	2823	89	3037	99
8	2618	64	2794	74	3063	95	2424	76	2405	75	2433	83	2615	77	2633	82	2888	111
9	2737	66	2937	76	3021	101	2711	71	2554	80	2604	104	2623	70	2813	81	3068	100
10	2713	69	2792	83	2897	105	2384	78	2341	81	2522	94	2699	78	2695	83	3008	105

Table B.26 The results of the MRHP_SR for the 20x15x[16,20] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_SR 5 window	
1	2209	186	2136	230	2721	280
2	2173	177	2335	218	2624	260
3	2377	170	2385	185	2911	271
4	2090	180	2277	219	2696	245
5	2153	169	2328	210	2481	259
6	1993	167	2161	195	2531	285
7	2148	180	2176	197	2792	263
8	2166	178	2365	212	2661	299
9	2217	174	2256	212	2672	278
10	2212	196	2225	234	2636	342

Table B.27 The results of the MRHP methods for the 20x15x[16,20] problems for special reentrant job shop

prb	MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_FJOB 5 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BMCH 5window		MRHP_BJOB 2 window		MRHP_BJOB 3 window		MRHP_BJOB 5 window	
1	2935	177	3173	213	3285	304	2764	189	2775	221	3255	251	2941	206	2952	216	3170	261
2	2796	192	3044	216	3179	293	2503	207	2468	231	3039	288	2894	221	2828	209	2998	309
3	3030	184	3183	216	3329	309	2657	193	2905	206	3027	260	2837	199	3093	229	3309	262
4	2687	171	3005	208	3036	315	2402	170	2571	205	2795	250	2613	182	2733	214	3162	278
5	2904	202	3186	204	3182	319	2596	191	2957	225	2726	247	2907	212	2895	264	3054	252
6	2939	177	3028	190	3194	265	2593	190	2706	220	2865	218	2815	178	2905	226	3123	252
7	2919	182	3102	222	3118	310	2600	195	2554	196	2582	270	2877	191	2942	229	3028	259
8	2858	180	3133	220	3177	303	2603	198	2529	224	2997	273	2835	180	3025	226	3139	276
9	2831	186	3076	205	3133	264	2548	201	2817	230	2906	236	2810	218	2782	196	3049	233
10	2824	189	2989	208	3130	332	2716	199	2779	242	2750	269	2844	204	2900	251	3186	281

Table B.28 The results of the MRHP_SR for the 20x20x[21,25] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_SR 5 window	
1	2447	418	2535	504	3064	686
2	2518	431	2886	487	3277	664
3	2356	421	2562	481	2812	563
4	2405	408	2668	471	3282	638
5	2429	409	2648	489	3268	604
6	2326	376	2442	471	3043	653
7	2345	386	2785	479	2994	596
8	2581	422	2826	482	3011	609
9	2360	424	2486	499	3092	625
10	2528	397	2593	468	3094	627

Table B.29 The results of the MRHP methods for the 20x20x[21,25] problems for special reentrant job shop

prb	MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_FJOB 5 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BMCH 5window		MRHP_BJOB 2 window		MRHP_BJOB 3 window		MRHP_BJOB 5 window	
1	2944	417	3107	523	3298	761	2692	462	3100	576	3281	650	2952	480	3103	572	3175	714
2	3157	472	3290	519	3498	774	2671	423	3092	491	3251	614	3053	483	3285	556	3341	674
3	2741	412	2933	516	3158	646	2637	434	2692	510	2899	600	2939	439	2912	530	3051	620
4	2899	382	3015	482	3084	635	2643	416	2761	478	3185	598	2855	416	3099	531	3039	596
5	2969	403	3133	528	3349	663	2731	424	2956	493	3151	563	2959	452	3162	571	3162	603
6	2863	405	3019	499	3100	711	2481	406	2786	474	3376	662	2842	409	3065	538	3143	622
7	3136	420	3112	506	3292	640	2776	428	3043	502	3229	593	2907	441	3183	485	3196	591
8	3018	391	3348	530	3276	709	2742	410	2986	506	3105	579	3046	396	3032	519	3153	598
9	2825	403	2975	487	3171	682	2673	425	2697	546	3239	677	2794	448	2958	476	3003	577
10	3078	373	3189	476	3251	658	2675	419	2841	510	3587	663	3051	432	3194	532	3174	628

Table B.30 The results of the MRHP_SR for the 30x10x[11,15] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_SR 5 window	
1	2447	418	2535	504	3156	183
2	2518	431	2886	487	2924	179
3	2356	421	2562	481	2840	176
4	2405	408	2668	471	3223	177
5	2500	400	2648	489	2830	168
6	2326	376	2442	471	3012	181
7	2345	386	2785	479	2873	176
8	2581	422	2826	482	2872	197
9	2366	424	2486	499	2934	185
10	2528	397	2593	468	3464	188

Table B.31 The results of the MRHP methods for the 30x10x[11,15] problems for special reentrant job shop

prb	MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_FJOB 5 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BMCH 5window		MRHP_BJOB 2 window		MRHP_BJOB 3 window		MRHP_BJOB 5 window	
1	4001	115	4127	179	4448	181	3748	128	3927	148	3885	176	3973	153	4026	156	4446	175
2	4318	122	4290	168	4749	193	3734	132	3858	147	3973	179	4001	159	4148	149	4400	181
3	4143	119	4336	147	4521	179	3541	133	3935	148	3886	174	4023	153	4080	144	4412	193
4	4073	121	4270	155	4438	199	3800	140	3360	150	3875	178	3931	166	4047	150	4302	198
5	3964	121	4076	160	4390	200	3564	139	3426	149	3754	178	3777	180	3898	151	4261	179
6	3934	117	4185	163	4350	185	3683	136	3812	147	3947	161	3960	133	4017	156	4697	175
7	4009	116	4116	142	4292	189	3457	124	3562	151	3655	199	3811	128	3977	146	4262	169
8	3961	123	4138	149	4319	219	3566	140	3903	168	3888	178	3490	140	4007	152	4247	193
9	4200	114	4295	139	4703	205	3751	123	3835	151	3858	171	4032	125	4379	154	4591	189
10	4225	128	4395	155	4627	229	4047	150	3877	179	4035	190	3745	143	4151	169	4404	185

Table B.32 The results of the MRHP_SR for the 30x15x[16,20] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_SR 5 window	
1	3090	364	3219	412	3231	534
2	2903	335	2826	384	3097	555
3	2944	335	2920	402	3297	546
4	2893	362	3116	416	3190	531
5	2870	338	2976	400	3222	535
6	2788	328	3009	405	3264	490
7	3019	365	2999	395	3239	540
8	2796	369	3011	407	3289	547
9	2930	353	2941	421	3233	536
10	3072	381	3086	418	3264	541

Table B.33 The results of the MRHP methods for the 30x15x[16,20] problems for special reentrant job shop

prb	MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_FJOB 5 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BMCH 5window		MRHP_BJOB 2 window		MRHP_BJOB 3 window		MRHP_BJOB 5 window	
1	4064	355	4436	459	4612	682	3715	364	4200	433	4167	506	4034	412	4298	431	4468	600
2	4033	345	4427	442	4408	591	3682	367	3724	425	3918	489	4018	417	4235	471	4341	525
3	4124	359	4420	450	4563	693	3600	359	4038	433	4128	555	4055	392	4261	528	4493	571
4	4109	337	4454	421	4629	607	3646	372	3826	430	3939	542	4103	426	4201	464	4548	538
5	4093	362	4550	439	4613	642	3795	396	3757	436	3944	573	4064	411	4236	460	4522	593
6	3920	349	4342	397	4601	586	3582	374	3613	440	3921	518	3876	380	4186	466	4469	592
7	4161	359	4539	462	4611	664	3708	403	3803	404	3939	542	4148	402	4356	457	4732	636
8	4004	356	4295	449	4463	647	3649	385	4048	397	3798	527	4072	388	4078	448	4395	606
9	4056	357	4400	461	4603	665	3755	448	3438	483	3774	582	3987	429	4139	450	4331	609
10	4077	375	4407	441	4545	691	3769	408	3983	442	3869	591	4007	421	4364	541	4398	657

Table B.34 The results of the MRHP_SR for the 30x20x[21,25] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_SR 5 window	
1	3141	821	3505	1004	3832	1349
2	3295	840	3357	992	3996	1343
3	3218	882	3404	1000	3902	1280
4	3336	910	3402	961	4120	1304
5	3145	829	3158	985	3720	1297
6	3323	879	3294	952	3996	1252
7	3160	872	3520	976	4014	1238
8	3512	938	3549	1023	4270	1257
9	3007	940	3075	1028	3539	1264
10	3046	911	3115	948	3631	1220

Table B.35 The results of the MRHP methods for the 30x20x[21,25] problems for special reentrant job shop

prb	MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_FJOB 5 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BMCH 5window		MRHP_BJOB 2 window		MRHP_BJOB 3 window		MRHP_BJOB 5 window	
1	4380	836	4530	1009	4728	1551	3864	876	4257	1061	4520	1376	4269	953	4473	1174	4606	1284
2	4343	828	4514	987	4943	1528	3860	917	4056	1124	4325	1166	4245	978	4467	1002	4691	1263
3	4146	832	4557	1008	4548	1526	3655	869	4092	1116	3898	1344	4191	1049	4213	963	4401	1313
4	4239	830	4598	1018	4805	1394	3763	892	4155	1112	4703	1426	4180	919	4460	1064	4700	1229
5	4207	823	4600	1009	4721	1535	3961	1025	4184	1144	4215	1275	4334	937	4350	980	4702	1273
6	4300	815	4536	1015	4596	1351	3756	920	3913	999	4395	1205	4080	1057	4382	1008	4525	1182
7	4280	833	4553	992	4636	1412	3733	881	3773	1083	4448	1354	4139	901	4383	1034	4610	1288
8	4479	866	4556	1019	4895	1516	3882	828	4518	1107	4776	1325	4377	884	4560	1116	4780	1353
9	4200	851	4441	1021	4770	1441	3846	887	4128	1081	4263	1261	4171	919	4180	1112	4502	1190
10	4190	931	4449	1048	4623	1403	3697	922	4210	986	4275	1379	4156	893	4300	1000	4565	1492

Table B.36 The results of the MRHP_SR for the 50x10x[11,15] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_SR 5 window	
1	4133	295	4220	336	4368	428
2	4289	295	4272	352	4479	414
3	4008	288	4376	350	6382	408
4	4122	283	4171	334	4264	401
5	3967	355	4224	360	4383	416
6	4025	422	4219	335	6388	414
7	4079	315	3929	351	3998	416
8	4091	325	4251	314	4479	400
9	4032	336	4366	341	4155	438
10	4110	290	4183	318	4079	377

Table B.37 The results of the MRHP methods for the 50x10x[11,15] problems for special reentrant job shop

prb	MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_FJOB 5 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BMCH 5window		MRHP_BJOB 2 window		MRHP_BJOB 3 window		MRHP_BJOB 5 window	
1	6774	287	7185	334	7455	513	6039	382	6109	358	6070	460	6568	415	6832	376	7278	457
2	6814	286	7031	339	7485	487	6119	352	6522	349	6398	447	6492	388	6813	682	7452	465
3	6701	286	6769	336	7288	483	5665	325	6073	343	5967	460	6424	364	6718	392	7230	460
4	6689	292	7053	329	7503	503	5832	332	6121	503	6102	455	6240	318	6842	403	7244	476
5	6887	303	7050	341	7334	492	6235	344	5611	451	6239	462	6540	390	6535	430	7366	465
6	6603	288	7050	335	7395	759	6138	347	6309	360	6361	473	6417	483	6724	427	7235	465
7	6302	291	6700	335	7115	491	5446	315	5804	360	5564	444	5936	411	6236	353	7239	471
8	6948	288	6998	331	7889	531	6168	338	6359	369	6068	458	6370	390	6994	364	7491	470
9	6746	282	7094	343	7539	535	5875	340	6371	378	6224	495	6375	398	6627	383	7087	511
10	6493	285	6767	310	7247	452	6000	300	5509	345	5697	442	6374	397	6392	345	6749	444

Table B.38 The results of the MRHP_SR for the 50x20x[21,25] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_SR 5 window	
1	4560	2158	4748	2454	5183	3243
2	4418	2058	4641	2326	4940	3493
3	4580	2104	4680	2517	4970	3108
4	4477	2102	4589	2761	5140	3265
5	4943	2186	4650	2505	5361	3293
6	4562	2154	4540	2399	5212	3469
7	4558	2157	4646	2501	5261	3126
8	4404	2208	4657	2361	5132	3261
9	4544	2281	4621	2731	5019	3110
10	4564	2155	4714	2482	5133	3263

Table B.39 The results of the MRHP methods for the 50x20x[21,25] problems for special reentrant job shop

prb	MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_FJOB 5 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BMCH 5window		MRHP_BJOB 2 window		MRHP_BJOB 3 window		MRHP_BJOB 5 window	
1	6929	2307	7373	2570	7560	3952	5964	2470	6285	2763	6272	3690	6828	2292	7020	3097	7500	3582
2	6903	2299	7064	2809	7334	3812	5528	2221	6189	2890	6076	3380	6533	2334	7072	3148	7495	3578
3	7086	2286	7266	2637	7702	4042	5865	2348	6630	2392	6532	3380	6779	2504	7306	2818	7603	3668
4	6747	2247	7217	2877	7564	3956	5888	2262	6309	2762	6331	3503	6658	2869	6745	3271	7204	3533
5	7080	2487	7382	2707	7566	3954	6075	2282	6333	2736	6565	3873	7044	2448	7212	3221	7798	3386
6	6761	2185	7406	2565	7608	4085	5869	2352	6120	2707	6344	3407	6714	2422	6934	3257	7423	3702
7	7264	2412	7445	2589	7787	4131	6056	2464	6640	2701	6571	3607	6907	2280	7216	3225	7788	3586
8	6689	2121	7268	2637	7174	3778	5867	2350	5840	2883	6071	3417	6524	2292	6809	2942	7503	3580
9	6883	2211	7294	2528	8005	3929	5938	2340	6469	2894	6705	3750	6749	2385	6966	3093	7642	3511
10	6695	2443	6962	2455	7334	3905	5622	2413	6024	2784	6385	3556	6408	2401	6944	2794	7037	3678

Table B.40 The results of the MRHP_SR for the 100x10x[11,15] problems for special reentrant job shop

prb	MRHP_SR 2 window		MRHP_SR 3 window		MRHP_SR 5 window	
1	7766	1127	7840	1237	7766	1408
2	7695	1092	7860	1485	8034	1499
3	7771	1033	7967	1169	8163	1474
4	7687	1089	7831	1243	7860	1484
5	7481	1069	7858	1481	7859	1483
6	7466	1120	7578	1209	7564	1542
7	7997	1108	7987	1249	7858	1482
8	7669	1081	7855	1482	7771	1493
9	7693	1086	7859	1484	7862	1486
10	7689	1089	7864	1483	7856	1480

Table B.41 The results of the MRHP methods for the 100x10x[11,15] problems for special reentrant job shop

prb	MRHP_FJOB 2 window		MRHP_FJOB 3 window		MRHP_FJOB 5 window		MRHP_BMCH 2 window		MRHP_BMCH 3 window		MRHP_BMCH 5window		MRHP_BJOB 2 window		MRHP_BJOB 3 window		MRHP_BJOB 5 window	
1	13441	1122	13780	1313	14932	1876	11758	1295	12255	1384	12089	1745	12152	1519	13438	1809	13943	1757
2	13510	1167	14203	1281	14850	2134	12014	1262	12701	1551	12356	1708	12832	1435	13311	1497	14360	1981
3	13721	1100	13785	1299	14944	1873	11376	1219	12438	1310	12259	1837	12486	1418	13380	1484	14274	1794
4	13509	1167	13667	1333	14646	1868	11345	1241	12300	1557	12097	1772	12385	1539	13215	1608	14167	1699
5	13514	1167	13201	1357	14068	1875	11122	1257	12342	1420	11566	1967	11551	1482	12215	1489	13303	1941
6	13734	1129	13776	1307	14703	1817	11747	1308	12026	1411	12394	1811	12143	1520	13213	1604	14131	1884
7	13357	1260	13859	1376	14765	2035	11878	1216	12522	1342	12031	1693	12722	1460	13316	1723	13989	1621
8	13285	1103	13778	1310	14594	1897	11460	1234	12452	1340	12102	1879	12677	1475	13322	1449	14334	2024
9	13634	1390	13913	1287	14696	1931	11561	1249	12130	1428	12589	1806	12332	1498	13504	1796	14065	1838
10	13408	1066	13819	1243	14764	2005	11352	1209	12255	1463	12115	1569	12324	1423	13208	1606	14059	1836

APPENDIX C

Data For The Application Problem

Table C.1 The machine sequence of day 13

job	operation	machine	machine	fabric length	processing time on
1	1	1	80	2500	31
1	2	2	60	2500	42
1	3	4	90	2500	28
1	4	6	40	2500	63
1	5	3	90	2500	28
1	6	4	90	2500	28
1	7	5	100	2500	25
1	8	8	100	2500	25
1	9	17	100	2500	25
1	10	18	45	2500	56
1	11	19	60	2500	42
1	12	20	30	2500	83
2	1	1	80	900	11
2	2	2	60	900	15
2	3	4	90	900	10
2	4	17	100	900	9
2	5	8	100	900	9
2	6	3	90	900	10
2	7	17	100	900	9
2	8	19	60	900	15
2	9	20	30	900	30
3	1	1	80	450	6
3	2	2	60	450	8
3	3	4	90	450	5
3	4	8	100	450	5
3	5	3	90	450	5
3	6	8	100	450	5
3	7	17	100	450	5
3	8	18	45	450	10
3	9	19	60	450	8
3	10	20	30	450	15
4	1	1	80	350	4
4	2	2	60	350	6
4	3	3	90	350	4
4	4	4	90	350	4
4	5	5	100	350	4
4	6	8	100	350	4
4	7	5	100	350	4
4	8	8	100	350	4
4	9	15	200	350	2
4	10	17	100	350	4
4	11	18	45	350	8
4	12	19	60	350	6
4	13	20	30	350	12
5	1	1	80	600	8
5	2	2	60	600	10
5	3	4	90	600	7
5	4	8	100	600	6
5	5	3	90	600	7
5	6	8	100	600	6
5	7	17	100	600	6
5	8	18	45	600	13
5	9	19	60	600	10
5	10	20	30	600	20
6	1	1	80	1500	19
6	2	2	60	1500	25
6	3	3	90	1500	17

Table C.1 The machine sequence of day 13

job	operation	machine	machine	fabric length	processing time on
6	4	6	40	1500	38
6	5	8	100	1500	15
6	6	3	90	1500	17
6	7	8	100	1500	15
6	8	17	100	1500	15
6	9	18	45	1500	33
6	10	17	100	1500	15
6	11	19	60	1500	25
6	12	20	30	1500	50
7	1	1	80	500	6
7	2	2	60	500	8
7	3	4	90	500	6
7	4	6	40	500	13
7	5	3	90	500	6
7	6	8	100	500	5
7	7	17	100	500	5
7	8	18	45	500	11
7	9	19	60	500	8
7	10	20	30	500	17
8	1	1	80	650	8
8	2	2	60	650	11
8	3	4	90	650	7
8	4	6	40	650	16
8	5	3	90	650	7
8	6	8	100	650	7
8	7	17	100	650	7
8	8	18	45	650	14
8	9	19	60	650	11
8	10	20	30	650	22
9	1	1	80	3000	38
9	2	2	60	3000	50
9	3	4	90	3000	33
9	4	7	35	3000	86
9	5	3	90	3000	33
9	6	7	35	3000	86
9	7	17	100	3000	30
9	8	18	45	3000	67
9	9	19	60	3000	50
9	10	20	30	3000	100
10	1	1	80	2950	37
10	2	2	60	2950	49
10	3	4	90	2950	33
10	4	6	40	2950	74
10	5	3	90	2950	33
10	6	4	90	2950	33
10	7	5	100	2950	30
10	8	8	100	2950	30
10	9	17	100	2950	30
10	10	18	45	2950	66
10	11	19	60	2950	49
10	12	20	30	2950	98
11	1	1	80	1600	20
11	2	2	60	1600	27
11	3	3	90	1600	18
11	4	6	40	1600	40
11	5	3	90	1600	18
11	6	4	90	1600	18
11	7	5	100	1600	16
11	8	8	100	1600	16
11	9	17	100	1600	16
11	10	18	45	1600	36
11	11	19	60	1600	27
11	12	20	30	1600	53

Table C.1 The machine sequence of day 13

job	operation	machine	machine	fabric length	processing time on
12	1	1	80	1000	13
12	2	2	60	1000	17
12	3	4	90	1000	11
12	4	6	40	1000	25
12	5	3	90	1000	11
12	6	17	100	1000	10
12	7	11	200	1000	5
12	8	12	80	1000	13
12	9	5	100	1000	10
12	10	8	100	1000	10
12	11	14	200	1000	5
12	12	16	60	1000	17
12	13	12	80	1000	13
12	14	14	200	1000	5
12	15	17	100	1000	10
12	16	19	60	1000	17
12	17	20	30	1000	33
13	1	1	80	1000	13
13	2	2	60	1000	17
13	3	4	90	1000	11
13	4	6	40	1000	25
13	5	3	90	1000	11
13	6	17	100	1000	10
13	7	11	200	1000	5
13	8	12	80	1000	13
13	9	5	100	1000	10
13	10	8	100	1000	10
13	11	14	200	1000	5
13	12	16	60	1000	17
13	13	12	80	1000	13
13	14	14	200	1000	5
13	15	17	100	1000	10
13	16	19	60	1000	17
13	17	20	30	1000	33
14	1	1	80	550	7
14	2	2	60	550	9
14	3	17	100	550	6
14	4	11	200	550	3
14	5	12	80	550	7
14	6	5	100	550	6
14	7	7	35	550	16
14	8	14	200	550	3
14	9	17	100	550	6
14	10	19	60	550	9
14	11	20	30	550	18
15	1	1	80	1050	13
15	2	2	60	1050	18
15	3	17	100	1050	11
15	4	11	200	1050	5
15	5	12	80	1050	13
15	6	5	100	1050	11
15	7	7	35	1050	30
15	8	14	200	1050	5
15	9	10	35	1050	30
15	10	17	100	1050	11
15	11	19	60	1050	18
15	12	20	30	1050	35
16	1	1	80	1500	19
16	2	2	60	1500	25
16	3	17	100	1500	15
16	4	11	200	1500	8
16	5	12	80	1500	19
16	6	5	100	1500	15

Table C.1 The machine sequence of day 13

job	operation	machine	machine	fabric length	processing time on
16	7	7	35	1500	43
16	8	14	200	1500	8
16	9	17	100	1500	15
16	10	19	60	1500	25
16	11	20	30	1500	50
17	1	1	80	2200	28
17	2	2	60	2200	37
17	3	4	90	2200	24
17	4	5	100	2200	22
17	5	3	90	2200	24
17	6	4	90	2200	24
17	7	5	100	2200	22
17	8	8	100	2200	22
17	9	17	100	2200	22
17	10	18	45	2200	49
17	11	19	60	2200	37
17	12	20	30	2200	73
18	1	1	80	1950	24
18	2	2	60	1950	33
18	3	3	90	1950	22
18	4	6	40	1950	49
18	5	3	90	1950	22
18	6	8	100	1950	20
18	7	17	100	1950	20
18	8	18	45	1950	43
18	9	19	60	1950	33
18	10	20	30	1950	65
19	1	1	80	950	12
19	2	2	60	950	16
19	3	3	90	950	11
19	4	6	40	950	24
19	5	3	90	950	11
19	6	4	90	950	11
19	7	5	100	950	10
19	8	17	100	950	10
19	9	15	200	950	5
19	10	10	35	950	27
19	11	17	100	950	10
19	12	19	60	950	16
19	13	20	30	950	32
20	1	1	80	1500	19
20	2	2	60	1500	25
20	3	3	90	1500	17
20	4	17	100	1500	15
20	5	6	40	1500	38
20	6	4	90	1500	17
20	7	5	100	1500	15
20	8	3	90	1500	17
20	9	10	35	1500	43
20	10	13	35	1500	43
20	11	14	200	1500	8
20	12	13	35	1500	43
20	13	14	200	1500	8
20	14	17	100	1500	15
20	15	19	60	1500	25
20	16	20	30	1500	50