SOLUTION APPROACHES FOR FLEXIBLE JOB SHOP SCHEDULING PROBLEMS

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

BY

ŞERİFE AYTUĞ BALCİ

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

JAN 2013

Approval of the thesis:

SOLUTION APPROACHES FOR FLEXIBLE JOB SHOP SCHEDULING PROBLEMS

submitted by **ŞERİFE AYTUĞ BALCİ** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ___________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Murat Köksalan ___________________
Head of Department, **Industrial Engineering**

Prof. Dr. Meral Azizoğlu ___________________
Supervisor, **Industrial Engineering Dept., METU**

Dr. Cemal Berk Oğuzsoy ___________________
Co-Supervisor, **Roketsan Missile Industries Inc.**

Examining Committee Members:

Prof. Dr. Ömer Kırca ___________________
Industrial Engineering Dept., METU

Prof. Dr. Meral Azizoğlu ___________________
Industrial Engineering Dept., METU

Assoc. Prof. Ferda Can Çetinkaya ___________________
Industrial Engineering Dept., Cankaya University

Assist. Prof. Sakine Batun ___________________
Industrial Engineering Dept., METU

Dr. Cemal Berk Oğuzsoy ___________________
Planning, Strategy and IT Director,
Roketsan Missile Industries Inc.

Date: Jan 31, 2013

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

Name, Last name: Şerife Aytuğ Balci

Signature :

ABSTRACT

SOLUTION APPROACHES FOR FLEXIBLE JOB SHOP SCHEDULING PROBLEMS

Balci, Şerife Aytuğ
M.Sc., Department of Industrial Engineering
Supervisor : Prof. Dr. Meral Azizoğlu
Co-Supervisor : Dr. Cemal Berk Oğuzsoy

Jan 2013, 96 pages

In this thesis, we consider a flexible job shop scheduling problem existing in discrete parts manufacturing industries. We are motivated by the production environment of Roketsan Missiles Industries Incorporation, operating at Turkish defense industry. Our objective is to minimize the total weighted completion times of the jobs in the system.

We formulate the problem as a mixed integer linear program and find that our model could find optimal solutions only to small sized problem instances. For medium and large sized problem instances, we develop heuristic algorithms with high quality approximate solutions in reasonable solution time.

Our proposed heuristic algorithm has hierarchical approach and benefits from optimization models and priority rules. We improve the heuristic method via best move with non-blocking strategy and design several experiments to test the performances. Our computational results have revealed that proposed heuristic algorithm can find high quality solutions to large sized instances very quickly.

Keywords: Flexible Job Shops, Total Weighted Completion Time, Heuristic Approaches

ÖZ

ESNEK TİPLİ ATÖLYE ÇİZELGELEME PROBLEMLERİ İÇİN ÇÖZÜM YAKLAŞIMLAR

Balci, Şerife Aytuğ
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Meral Azizoğlu
Ortak Tez Yöneticisi : Dr. Cemal Berk Oğuzsoy

Ocak 2013, 96 Sayfa

Bu çalışmada, kesikli üretim sistemlerinde karşılaşılan Esnek Tipli Atölye Çizelgeleme Problemi ele alınmıştır. Türk savunma sanayisinde faaliyet gösteren Roketsan Roket Sanayi ve Ticaret A.Ş. tarafından motive edildik. Amacımız,toplam ağırlandırılmış iş bitiş sürelerini enazlamaktır.

Problemi karışık tamsayılı dorusal problem olarak formüle ettik ve modelin sadece küçük ölçekli problem örnekleri için optimal sonuç verebildiğini bulduk. Orta ve büyük ölçekli problem örnekleri için, makul çözüm süresinde yüksek kaliteye sahip yaklaşık çözümler veren sezgisel yöntemler geliştirdik.

Önerdiğimiz sezgisel algoritma hiyerarşik yaklaşıma sahiptir ve optimizasyon modelleri ve öncelik kurallarından yararlanmaktadır. Sezgisel metodu en iyi hareket yolu ve blokları kaldırma stratejisi ile geliştirdik ve performansını test etmek için birçok deney tasarladık. Deneylerimizin sonuçları önerdiğimiz sezgisel algoritmanın kısa sürede yüksek kaliteli sonuçlar verdiğini göstermiştir.

Anahtar kelimeler: Esnek Tipli Atölyeler, Toplam Ağırlıklandırılmış Bitiş Zamanları, Sezgisel Yaklaşımlar

To My Family and Friends

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my supervisor, Prof. Dr. Meral Azizoğlu for her guidance and encouragement throughout this study.

I would like to show my appreciation to my co-supervisor, Dr. Cemal Berk Oğuzsoy, for his precious discussions and comments during the progress of this thesis.

I thank to my family for their great support, encouragement and unshakable faith in me.

I would like to thank to my friends Kaan Balkan, Gizem Özden and Nebahat Kurt for their encouragement and motivation during this study.

For crucial help about the motivating company's production system, I would like to thank my friends Murat İleri, Emre Aytaç, Ufuk Çetin and Yasemin Arslan.

Furthermore, I want to thank my manager Mr. Kerim İnce who has been very supportive and encouraged me every step of the way.

I owe special thanks to Mr. Selçuk Yaşar, CEO of Roketsan Missile Industries Inc., for his great support to academic studies in the company.

Roketsan Missile Industries Inc. is gratefully acknowledged for this study.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: LITERATURE SURVEY 5

2.1. SOLUTION APPROACHES 5
2.1.1. OPTIMIZATION TECHNIQUES 5
2.1.2. HEURISTIC METHODS 6
2.1.2.1. MACHINE ASSIGNMENT RULES 6
2.1.2.2. OPERATION SEQUENCING (DISPATCHING) RULES 7
2.1.2.3. LOCAL (NEIGHBORHOOD) SEARCH METHODS 7
2.2. LITERATURE REVIEW 9
2.2.1. SINGLE OBJECTIVE (CRITERION) PROBLEMS 9
2.2.2. MULTI CRITERIA OPTIMIZATION 12

CHAPTER 3: AN APPLICATION: ROKETSAN CASE 15

3.1. GENERAL INFORMATION ABOUT ROKETSAN 15
3.2. THE MANUFACTURING SYSTEM 16
3.3. THE PRODUCTION PLAN AND MATERIAL FLOW 18
3.4. SCHEDULING DECISIONS 20

CHAPTER 4: A MIXED INTEGER LINEAR PROGRAMMING FORMULATION OF THE FLEXIBLE JOB SHOP SCHEDULING PROBLEM 21

CHAPTER 5: SOLUTION APPROACHES 30

5.1. CONSTRUCTION PHASE 30
5.1.1. ALLOCATION DECISIONS 31
5.1.1.1. MATHEMATICAL MODEL 31
5.1.1.2. SPT AND IMPROVED SPT ALLOCATION RULE 33
5.1.2. SEQUENCING DECISIONS 34
5.1.2.1. MATHEMATICAL MODEL (MILP-S) 34
5.1.2.2. PRIORITY RULES 35
5.2. IMPROVEMENT PHASE 35

CHAPTER 6: AN ILLUSTRATIVE EXAMPLE FROM ROKETSAN COMPANY............ 38

6.1. MILP SOLUTIONS.. 42
6.2. HEURISTIC SOLUTIONS 43
6.3. SCHEDULE IMPLEMENTED IN ROKETSAN 51
6.4. EVALUATION OF ALL SOLUTIONS 52

CHAPTER 7: COMPUTATIONAL RESULTS 57

7.1 PERFORMANCE OF THE MIXED INTEGER LINEAR MODEL 57
7.1.1 PURE OPTIMIZATION (FINDING OPTIMAL SOLUTIONS) 58
7.1.2 OPTIMIZATION MODEL WITH GAP 0.1 (FINDING UPPER BOUNDS) 59
7.2 PERFORMANCE OF THE HEURISTIC PROCEDURES.. 60
7.2.1 PRELIMINARY RUNS 60
7.2.2 MAIN RUNS.. 67
7.3 SUMMARY OF COMPUTATIONAL EXPERIMENTS 72
7.4 A GUIDE FOR THE SELECTION OF HEURISTICS 73

CHAPTER 8: CONCLUSIONS AND FURTHER RESEARCH DIRECTIONS.............. 76

APPENDICES 78

REFERENCES..104

LIST OF TABLES

Table 1: One-Week Production System Instance 37
Table 2: Machine Availability Times 40
Table 3: Jobs' Ready Times 40
Table 4: The sequence returned by the MILP at the end of 4 hours 41
Table 5: Allocation Results 42
Table 6: The Sequence Returned By MILP-S with Assignment Model 43
Table 7: The Sequence Returned By MILP-S with ISPT Rule 44
Table 8: The Sequence Returned By Priority Rule 3 with Assignment Model..... 45
Table 9: The Sequence Returned By Priority Rule 3 with ISPT Rule 46
Table 10: The Sequence Returned From Improvement Phase with Assignment Model (Best Improving Strategy) 47
Table 11: The Sequence Returned From Improvement Phase with ISPT Rule (Best Improving Strategy) 48
Table 12: The Sequence Returned From Improvement Phase with Assignment Model (Best Move with Non-Blocking Strategy) 49
Table 13: The Sequence Returned From Improvement Phase with ISPT Rule (Best Move with Non-Blocking Strategy) 50
Table 14: Real Schedule for Week 3 51
Table 15: Completion Times of Jobs Returned By the Solution Approaches...... 52
Table 16: Total Weighted Flow Time Values of the Methods -- Week 3............ 53
Table 17: Total Weighted Flow Time Values of the Methods -- Week 1............ 54
Table 18: Total Weighted Flow Time Values of the Methods -- Week 2............ 55
Table 19: Total Weighted Flow Time Values of the Methods -- Week 4............ 55
Table 20: Parameter Values Used at Instances 56
Table 21: The Problem Sizes Used in Our Experiments 57
Table 22: The CPU Times and Number of Unsolved Instances in 1 hour 58
Table 23: The CPU Times and Number of Unsolved Instances with 10% gap ... 59
Table 24: The Deviations of 10% gap from the Optimal 60
Table 25: Number of Instances the Assignment Rule's Machine Loads Same with Optimal Solution's Machine Loads 61
Table 26: Number of Optimal Machine Assignments with Allocation Rules 61

Table 27: Percentage of Optimal Machine Assignments with Allocation Rules.....62
Table 28: Percent Deviation of the Heuristic Procedures from the Optimal Solutions 64
Table 29: The CPU times (seconds) of the Heuristic Procedures 66
Table 30: Number of Optimal Solution Given By MILP-S Model 67
Table 31: The Number of Best Solution by the Heuristic Methods................... 68
Table 32: Percent Deviation of Heuristic Methods from the Best Value............ 68
Table 33: The CPU times of the Heuristic Methods 69
Table 34: Deviations for the Improvement Procedures-7 Jobs, 2-4 Operations 70
Table 35: CPU times for the Improvement Procedures-7 Jobs, 2-4 Operations. 71
Table 36: % Improvement in the Best Value over 100 iterations.................... 71
Table 37: Solution Methods and Resource Requirements as a Function of the Problem Size 73
Table 38: Week 1 Production System Instance 77
Table 39: Week1-Machine Availability Times 80
Table 40: Week 1-Jobs’ Ready Times 80
Table 41: Week 1-Jobs’ Weights 81
Table 42: Week1-The sequence returned by the MILP at the end of 4 hours.... 81
Table 43: Week1-Completion times of jobs for MILP model 82
Table 44: Week1-Allocation Results 82
Table 45: Week1-The Sequence Returned By MILP-S with Assignment Model .. 84
Table 46: Week1-The Sequence Returned By MILP-S with ISPT Rule 85
Table 47: Week1-Completion Times of Jobs Returned By MILP-S with ISPT Rule.. 86
Table 48: Completion Times of Jobs Returned By MILP-S with Assignment Model... 87
Table 49: Week 2 Production System Instance 88
Table 50: Week 2- Machine Availability Times 90
Table 51: Week 2- Jobs’ Ready Times 91
Table 52: Week2- Allocation Results 91
Table 53: Week 4 Production System Instance 93
Table 54: Week 4- Machine Availability Times 96
Table 55: Week 4- Jobs’ Ready Times 97
Table 56: Week 4- Jobs’ Weights 97
Table 57: Week 4-The sequence returned by the MILP at the end of 4 hours... 98
Table 58: Week 4-Completion times of jobs for MILP model 99
Table 59: Week4-Allocation Results..100

LIST OF FIGURES

Figure 1: X-Axis Motion 15
Figure 2: Y-Axis Motion 15
Figure 3: Z-Axis Motion 15
Figure 4: C-Axis Motion 16
Figure 5: Workshop Layout 17
Figure 6: Production Flow Diagram 18
Figure 7: A Three Jobs-Three Machines Problem Instance 22
Figure 8: Example for neighborhood structure Se_1 35
Figure 9 : Solution Procedure for FJSP Problem 36
Figure 10: The Average CPU Times of Optimal Solutions -2 and 4 Operations Case 58
Figure 11: The Average CPU Times of 10% Gap Solutions-2 and 4 Operations Case 59
Figure 12: CPU Times of Priority Rules........70

CHAPTER 1

INTRODUCTION

In today competitive business environment, the companies aim to increase their profits by producing high quality products at low cost and high speed. Investing in assets, e.g. labor, tooling, machine, material and equipment, and employing these assets efficiently to satisfy the customer needs is the only way to stay in the market in long term at high profit levels.

Operating the company assets efficiently in the short-term by prescribing when and where to perform each operation required to manufacture the product is called production scheduling. The main goal of production scheduling is to sequence operations on production resources and define their start time and completion times on each resource.

The production scheduling system of a company can be categorized as one-stage or multi-stage. If all jobs have a single operation, the production scheduling is referred to as one-stage; otherwise it is a multi-stage system. In one-stage problems all operations are performed by a single machine, hence there is no resource allocation problem. In multi-stage systems a route, that is a processing sequence of operations for each job, is specified.

According to the operation sequence of the jobs, the multi-stage production scheduling are categorized as flow shops, job shops and open shops. Assume a manufacturing environment with n jobs and m machines. In a flow shop environment each of the n jobs must be processed through the m machines in the same order, i.e., all jobs have identical routes and each job is processed exactly once on each machine. (Nahmias, 2005). A job shop scheduling differs from a flow shop in that not all jobs are assumed to require exactly m operations and some jobs may require multiple operations on a single machine. Furthermore, in a job shop each job may have a different required sequencing of operations, i.e., the routes are arbitrary (Nahmias, 2005). Open shop schedule is different from flow shop and job shop schedules as there are no restrictions placed on the sequence of the operations. The operations of the job can be processed at any order and finding the processing sequence is an additional problem.

The multi-stage scheduling also differs according to the machine flexibility for the operation to be performed. A machine may have the flexible capability to be set up to process more than one type of operations. (Fattahi, 2007).

For the long term survival in the rapidly changing market, the manufacturing environments have to become more adaptive, hence flexible, supported by the multi functional alternative machines to perform each operation. The flexible job shop scheduling problem (FJSP) is an extension of the job shop scheduling problem where operations are allowed to be processed on the multiple capable machines. The flexible job shop scheduling problem is formed by assigning each operation to a machine out of a set of capable machines and sequencing the assigned operations on each machine in order to obtain a feasible schedule by considering a predefined objective function.

FJSP problems assume that each machine can process only one operation at a given time. Each operation is assigned to one of its capable machines and is processed on its assigned machine without interruption. There are precedence constraints that indicate the processing order between operations of a job. The operation cannot be performed before all its predecessor operations are finished. In this study, we also use ready time constraints and initial machine available time constraints. The ready time constraints exist when the jobs enter into production system at different times from the start of the scheduling horizon due to their arbitrary delivery times from the suppliers or subparts of the job continuing to be processed in other production departments. The initial machine available time constraints exist as the machines may become available at different times due to the maintenances required to start the batch of the scheduling horizon or due to the operations pending from the previous scheduling horizon.

By considering all constraints, the FJSP problem finds the sequencing of the operations on each machine so as to optimize a specified performance measure. Makespan is the time needed to complete all jobs, and is most commonly used performance measure. In the literature some machine-related objective functions like minimizing total machine idle time, minimizing total workload and minimizing the maximum workload to allocate jobs to machines are also used. Minimizing total completion time (equivalently total flow time) and minimizing total weighted completion time (equivalently total weighted flow time) are two important objective functions in real world scheduling applications. Flow time of a job is the time it spends in the system hence is an indicator of the work in process inventory level. The weighted flow time reflects the relative importance of keeping in the inventory hence considers the value of the work in process inventory. The relative importance might also define the priority of the jobs' customers. The more prestigious customer would receive higher weight; hence his order would be delivered earlier under total weighted flow time objective. Despite their practical importance, on flexible job shops there is a unique study by Lee et al. (2012) that considers total flow time objective and no study on the total weighted completion time objective. Recognizing this gap in the literature, we consider total weighted completion time as the objective function of our flexible job shop scheduling problem. If the job has higher weight due to the customer or production needs, the model attempts to finish this job earlier. A higher weight may be used as an indicator of lower customer due date as well.

In the literature two types of the FJSP are mentioned according the speed of the capable machines. For type I FJSP, jobs can have alternative identical machines for each operation. The problem is to assign each operation to one of its identical capable machines and to find the sequence of operations on each machine. For type II FJSP, operations can have non-identical capable machines to be performed at. The problem is to assign each operation to one of its non-identical capable machines and arrange assigned operations at each machine.

In this study we consider the type II FJSP where the operations have non-identical machines to be performed at. We assume that there are at most two operations having the flexibility to exchange their places at the route. Hence we consider a special case of a mixed job where all operations, except two, follow job shop structure and two operations reflect open shop nature.

A practical application that we take our motivation from is a defense industry company, ROKETSAN Missiles Industries Incorporation. ROKETSAN is one of the defense castles of Turkey with rocket systems, air defense programs, antitank missiles, precision guided munitions and airbag productions; space and satellite projects, electronic warfare support center studies. Due to the characteristics of production, the company uses some special machines. The performance and capacity of entire system is limited by these special machines, hence to increase the performance of the overall production system, efficient use of the bottleneck machines is needed. The scheduling problem the company defines and thereafter we formalize in the study was assigning operations to one of its available machines and sequencing assigned operations at each machine at a week-period basis with the aim of finishing the jobs as early as possible. The schedule program takes production plan as an input and gives a weekly schedule for each machine as an output.

The flexible job-shop scheduling problem is a strongly NP-hard problem, since it is a generalization of the job shop scheduling problem that has been proven to be strongly NP-hard (Garey et al. 1976). Owing to this combinatorial complexity, all FJSPs propose polynomial time heuristic procedures to handle medium sized problem instances.

In this study, we first give the mathematical programming model of our problem and show that the model is capable of solving small sized problem instances. For medium and large sized problem instances we propose heuristic procedures that run in two steps: allocation and scheduling.

The rest of the thesis is organized as follows. In Chapter 2, we review the literature on the Flexible Job Shop Scheduling problems. In Chapter 3, we present the motivating case of the study. In Chapter 4, problem definition is given. In Chapter 5, we discuss solution approaches to find approximate solutions at different size problems. An illustrative example from the motivating company ROKETSAN is given in Chapter 6. We report the results of our computational experiments in Chapter 7. The study with our main findings and future research suggestions are concluded in Chapter 8.

CHAPTER 2

LITERATURE SURVEY

The flexible job-shop scheduling problem (FJSP) is an extension of the classical job shop scheduling problem (JSP) that assigns each operation to one of its capable machines and sequences the assigned operations on each machine.

In this chapter we first present the solution approaches used to solve the FJSP and then review the literature based on the objective functions used in the studies.

2.1. Solution Approaches

Garey et al. (1976) show that the job shop scheduling problem is strongly NP-hard, so is the flexible job shop scheduling problem.

For flexible job shop scheduling problems several different solution approaches have been proposed. The reported approaches can be classified into main groups as optimization methods and heuristic methods.

Due to the complexity of the problem, the optimization methods require enormous computational time to reach the optimal solution.
The solution times increase exponentially with the increases in the problem size, hence they are usually inapplicable to real life instances. The approximation methods aim to find a high quality solution in reasonable time. The returned solutions do not have any guarantee of optimality however have greater application chance due to their reasonable computational time requirements.

In the following, we briefly overview the most commonly used optimization and heuristic techniques to solve the FJSP.

2.1.1. Optimization Techniques

Mixed Integer Linear Programming

Mixed Integer Linear programming is an optimization technique used to solve many combinatorial optimization problems. The flexible job shop scheduling problems can be formulated as mixed integer linear programs; however their solutions are limited only to small-sized problem instances due to the exponential number of integer decision variables.

Implicit Enumeration Techniques

Branch and bound algorithms and dynamic programming algorithms are implicit enumeration techniques used for solving combinatorial optimization problems. Dynamic programming algorithms are not used to solve the FJSP due to their high memory requirements. Branch and bound techniques have an advantage of using less memory, however at an expense of high solution times. The technique partitions the problem into sub-problems and evaluates each sub-problem by lower and upper bounds. It guarantees optimality, generally runs quicker than the mathematical models, however still in exponential time.

2.1.2. Heuristic Methods

For the FSJP, the optimization methods cannot return solutions for the large-sized problem instances in reasonable times; hence the heuristic techniques that produce approximate solutions are needed. The time of reaching an approximate solution is usually much less than that of reaching the optimal solution. This is the fundamental basis of accepting the heuristic methods. Moreover in many real life FSJP instances, an optimal solution is not essential and a high quality approximate solution may be satisfactory. The emphasis is now placed by many researchers on developing heuristic algorithms that return solutions closest to the optimal solutions in the shortest possible time.
Detailed explanations of the most common heuristic techniques, machine assignment rules, dispatching rules and search procedures for the FJSS are given below.

2.1.2.1. Machine Assignment Rules

Machine assignment rules are simple procedures to assign an operation to one of its capable machines for the FJSP. They do not reside any iteration routine. Their choice basically depends on the objective function used.

Below is the most commonly used machine assignment rules excerpted from Urlings et al.(2010).

Random Assignment Rule: For every operation, a machine is selected randomly from a set of candidate machines. The main advantage is its computational simplicity whereas the disadvantage is the resulting poor machine workload balance.

Minimum Processing Time Rule: For every operation, the machine with minimum processing time in the candidate machine set is selected. This rule helps to reduce the total processing load over all machines and the completion time

Earliest Completion Time (ECT): For every operation, the machine that can complete the operation earliest is selected. The rule helps to reduce the makespan and total completion time.

Earliest Preparation Next Stage (EPNS): The machine that can prepare the job earliest to the next operation is chosen. Therefore time lags between the current and the next operation are taken into account. This rule uses more information about the continuation of the job than the previous rules, by focusing on the machines of the succeeding operations.

2.1.2.2. Operation Sequencing (Dispatching) Rules

Dispatching rules select a job from the set of jobs waiting for processing, whenever a machine is freed. The following is the most commonly used dispatching rules for the FSJP.

Service in Random Order (SIRO) Rule: Sequence the jobs randomly. No attempt is used to minimize any objective.

Job Slack (S): Give priority to the job with the least slack time where the slack time is the difference between the due date and remaining processing requirement. The rule tends to minimize the customer, usually lateness related objectives.

Earliest Due Date: Sequence the jobs according to their non decreasing order of due dates. The rule tends to minimize the customer, usually the lateness related objectives.

Earliest Release Date First (ERD) Rule: First-come job is processed first. This rule tends to minimize the variation in waiting time of the jobs.

Shortest Processing Time First (SPT) Rule: Process the jobs in non decreasing order of their total processing times. The rule tends to minimize the sum of total completion time of jobs.

Longest Processing Time First (LPT) Rule: Process the jobs in non increasing order of their total processing times. The rule tends to balance the workload over the machines.

Shortest Setup Time First (SST) Rule: Select the job with minimum setup time. This rule tends to minimize the total processing load.

Least Flexible Job First (LFJ) Rule: Select the job with minimum processing alternatives. The aim is to give a priority to the job having less chance for assignment, hence relaxing the future decisions.

2.1.2.3. Local (Neighborhood) Search Methods

Local search techniques attempt to find an acceptable near optimal solution in a large solution space generated by the FJSP. The quality of the algorithms is heavily dependent on the initial solutions and neighborhood structures. Heuristics or simple rules are used to get initial solutions. The neighborhood structure describes the relationship between neighbors. The search procedure starts with finding an initial solution and making small changes, so called moves, in the neighborhood of the current solution in an iterative manner. Each iteration makes an improvement on the objective function, so called fitness, value.
The main problem with simple search techniques is to become trapped in a local optimal solution in the neighborhood which may be too far from the optimal solution.

i. Random Search

Random search is a basic method which explores the search space by randomly selecting solutions and evaluating its fitness value. It never gets stuck in a local optimum as the next solution is selected randomly, not in a defined neighborhood. The procedure is likely to produce satisfactory solutions if a big number of iterations that explores the search space is performed. However, for most problems, exploring the whole search space requires too much computation effort and it is far from being practical.
We next discuss the solution approaches that select the moves from the current solution in a defined neighborhood, hence in an efficient way.

ii. Simulated Annealing

Threshold algorithms accept inferior schedules as possible moves if the deterioration in the objective function is less than some threshold value. Hence the threshold values determine whether a schedule can be disregarded or not. The simulated annealing algorithms use the idea of thresholds and vary the thresholds over the execution of the algorithms'. The basic steps in simulated annealing method can be stated as follows (Brandimarte, Villa, 1995).

1. Choose an initial solution and define a neighborhood structure.
2. Choose a new solution randomly in the neighborhood and compute the fitness value.
3. Calculate the acceptance probability (P).
4. Accept the new solution with probability of P. If the solution is accepted, replace with the old one.
5. If the termination condition is met, stop the algorithm, otherwise update the parameters and go to step 2.

iii. Tabu Search Algorithms

Tabu search is a local search technique that is characterized by its use of adaptive memory. The technique relies on the memory structures (describing the recently visited solutions, so called tabu moves) to avoid local minima and achieves an effective balance of intensification (reinforcing the attributes historically found good) and diversification (driving the search into new regions). At each iteration, the fitness value of each non tabu move is evaluated and the best move is accepted as the current solution. The mechanism used to escape from the local minima may lead to different problems. For example the increase in the size of the tabu list (that resides the tabu moves) increases the solution time and the decrease in the size of the tabu list may lead to an infinite loop, hence cannot lead to an improved solution. The tabu list size that well catches the tradeoff between the quality and speed of the solutions are generally based on empirical results.

iv. Genetic Algorithms

A genetic algorithm is a local search technique that mimics the natural process of evolution. The process analyses the fitness of a species in their particular environment. From one generation to the next, it preserves good characteristics and eliminates the poor ones. Analogous to this, genetic algorithms use strategies to decide which part of the solutions to retain and which ones to discard. These strategies are defined by a set of functions called genetic operators, the most common of which are selection, crossover and mutation. The aim of selection operator is to discriminate between good and bad schedules and produce more solutions from good solutions with favorable fitness values. The crossover operator takes two parents from the selection step and uses these solutions to generate new ones. The mutation operator tries to diversify the search by jumping into new neighborhoods and hence avoids for converging the local minima.

Genetic algorithms require a starting population, i.e., a set of initial solutions. These solutions can be generated randomly or produced intuitively by heuristics. Heuristic techniques use the specifics of the application, hence are likely to find more accurate solutions.

2.2. Literature Review

In this study we review the studies on the FJSP according to the objective functions.

2.2.1. Single Objective (Criterion) Problems

The majority of single criterion problems consider makespan as the objective function. To the best of our knowledge there is only one study that considers total completion time and one study that considers total workload of the machines. Below is detailed discussion of those studies.

i. Makespan

Bruker and Schlie (1990) are the first researchers that address the FJS makespan problem. They develop a polynomial algorithm for solving the flexible job shop problem with two jobs. For solving the realistic case with more than two jobs, hierarchical approaches and integrated approaches are constructed. In their hierarchical approach, the allocation problem and sequencing problem are treated separately and in their integrated approaches the allocation and sequencing problems are solved simultaneously.

Brandimarte (1993) uses TS algorithm to solve the FJSP in a hierarchical manner. He decomposes the problem into two sub problems as routing and scheduling problem which is obtained by assigning each operation of each job to one among the equivalent machines. Information flow between the two hierarchical levels is two-way. He makes experimentations with two types of initial solutions as the first one generated by priority rule of shortest processing time and as the second one generated by priority rule of the minimum total weighted tardiness. The results show that there is a great potential of improvement with respect to the results of priority rules.

Fattahi et al. (2007) develop a mathematical model and propose heuristic procedures. They report that their model is capable of solving small sized problem instances. Their heuristic procedures are of two types: Integrated and hierarchical. In integrated approaches machine assignment and sequencing decisions are given simultaneously whereas these decisions are sequential in hierarchical approaches. In both types they employ tabu search and simulated annealing (SA) heuristics. Their experimental results show the superiority of the hierarchical approach using the simulated annealing.

Xing et al. (2007) propose a multi-population interactive co-evolutionary algorithm. In the proposed algorithm, the ant colony optimization and genetic algorithm with different configurations are applied to evolve each population independently. The performance of the algorithm has been improved largely by integrating the ant colony optimization with the genetic algorithm and constructing the interaction, competition and sharing mechanism among populations. Using a set of benchmark instances taken from literature, they compare their approach with Temporal Decomposition, Controlled Genetic Algorithm, Approach by Localization, Approach by Localization combined with Controlled Genetic Algorithm, combination of Particle Swarm Optimization to assign operations on machines and Simulated Annealing to schedule operations and Tabu Search algorithm. The experimental results have shown that the proposed algorithm is a feasible and effective approach.

Zhang et al. (2008) propose a genetic algorithm combined with local search of TS algorithm. Time-varying crossover probability and time varying maximum step size of search are used to control the local search and convergence to the optimal solution. The performance of the proposed algorithm is tested on various benchmark problems and compared with the optimal makespan if known, otherwise, the best lower and upper bound found to date and Ho&Tay approach which is proposed by Ho and Tay. The results indicate that the proposed algorithm outperforms Ho's approach in all problems and it is effective and efficient when compared with optimal makespan.

Amiri et al. (2010) propose a variable neighborhood search (VNS) approaches. They present various neighborhood structures for machine selection and sequencing problem. They compare their findings with those of Kacem et al. (2002 a, 2002 b), Xia and Wu (2005), Zhang and Gen (2005) and Gen et al. (2008). Their results reveal that their algorithm performs compatible with the hybrid GA of Gen et al. and TS of Mastrollili and Gamberdella and better than the other methods compared.
Al-Hinai et al. (2010) propose hybridized GA architecture. By integrating it with an initial population generation algorithm and a local search method, the performance of the GA is improved. The proposed algorithm is compared with the GA proposed by Zribi et al. (2007), by Gao et al. (2008) and by Pezzella et al. (2008).The experimental results show that the proposed algorithm outperforms the GA proposed by Zribi et al. (2007) and the GA proposed by Pezzella et al. (2008) and produces a comparable quality to the algorithm proposed by Gao et al. (2008). The advantage of the proposed hybridized GA is that the genetic operators of crossover and mutation do not require a repair process to obtain a feasible schedule.

Li et al. (2011) present a novel hybrid TS algorithm with a fast public critical block neighborhood structure. They use three different approaches in the neighborhood structure for machine assignment and three insert and swap functions based on public critical blocks in the operation scheduling neighborhood structure. These approaches and functions are used to decrease the neighborhood size and eliminate unnecessary and infeasible moves. The proposed algorithm is compared with the AL+CGA approach proposed by Kacem et al. (2002), the PSO+SA presented by Xia et al. (2005), the PSO+TS developed by Zhang et al. (2009), the PVNS introduced by Yazdani et al. (2010), and the KBACO presented by Xing et al. (2010). The experimental results show that the proposed hybrid algorithm is competitive to the compared algorithms in terms of solution quality and computational efficiency.

Hama et al. (2011) present a real-time scheduling (RTS) heuristic model. The model selects and assigns a single job to the earliest available machines at each iteration. They formulated a binary IP model that includes machine compatibility constraints. They decompose the model into two phases: the first phase calculates a production target on each machine, and the second phase finds the schedule. By setting the production target by each machine found by the first model, the RTS model seeks the local optimality by only considering the current state. At each iteration, the RTS heuristic assigns one job to the subset of machine group until there is no remaining job. The proposed algorithm is compared with the exact optimal model and it gives sufficient results.
Chen et al. (2012) propose a genetic algorithm and a group genetic algorithm for the flexible job shop scheduling problem with reentrant process. Their procedure has two modules: machine selection and operation sequencing. Machine selection module selects a capable machine to each operation, and operation sequencing module finds the start time of each operation on its selected machine. They base their algorithm on a practical situation and show that their algorithm is superior to the existing methods.

Wang et al. (2012) propose an artificial bee colony algorithm which considers the balance between global exploration and local exploitation. The initial solutions with certain quality and diversity are generated using combined strategies. The crossover and mutation operators are used to generate the new neighbor food sources for the employed bees for machine assignment and operation sequencing. At last, a local search strategy based on critical path is developed in the searching framework. In addition, to transform a solution to an active schedule, a well-designed left-shift decoding scheme is employed. The satisfactory performance of the proposed algorithm is revealed by simulation results and comparisons with some existing algorithms, based on benchmark instances. The performance of ABC algorithm is tested by comparing it with several algorithms including AL + CGA (Kacem et al., 2002), GENACE (Tay et al., 2004), PSO + TS (Zhang et al., 2009), PVNS (Yazdani et al., 2012), KBACO (Xing et al., 2010) and TSPCB (Li et al., 2011). The proposed algorithm is competitive to the compared algorithms in terms of solution quality and computational efficiency.

ii. Total completion time:

To the best of our knowledge, there is only one study due to Lee et al. (2010) on minimizing total completion time objective.

Lee et al. (2010) solve the FJSP with AND/OR relations between the operations. OR precedence relations imply at least one of the specified operations should be performed, to start an operation. They develop a MILP that is capable of solving small-sized problem instances. For larger sized instances they propose local search heuristics namely tabu search (TS) and genetic algorithms (GA). Their experimental results show that their algorithms provide high quality solutions in reasonable time. The proposed methods are compared with those of SEA (symbiotic evolutionary algorithm) by Kim et al. (2003) and HA (hybrid algorithm) by Li et al. (2010). The results of the proposed methods were superior to those of both SEA and HA.

iii. Maximum Workload

The maximum workload among the machines objective is considered in a unique study by Ida et al. (2009). The study presents a new method of survival selection, a method of initial solution generation and mutation, and a method of escape to their genetic algorithm. They compare the proposed pGA with the algorithms proposed by Ho and Tay (GENACE) and Ong and colleagues (ClonaFLEX) in terms of early convergence. As indicated by the results, the performance of the proposed algorithm is equal to or better than the other algorithms except ClonaFLEX as it gives best. In order to verify the search performance of the proposed algorithm, they compare it with the algorithm proposed by Mastrolilli and Gambardella (TSopt) and pGA outperforms it in almost every case.

2.2.2. Multi Criteria Optimization

The multi criteria optimization studies for the flexible job shop scheduling problems consider any combination of makespan, total workload of the machines and maximum workload of the machines.

Kacem et al. (2002 a) propose a two-step approach for flexible job-shop scheduling problem with makespan and the total workload of the machines criteria. This multi-objective optimization is done in a suitable search space determined by an assignment algorithm. The first step is to apply the localization approach to solve the resource allocation problem and generate the assignment schemata. The second step is to apply a controlled evolutionary algorithm. The initial population constructs a starting point from the set of assignments found in the first stage. In such an approach, they apply advanced genetic manipulations in order to enhance the solution quality. They compare their proposed algorithm with the temporal decomposition, classic genetic algorithm and approach by localization. The proposed algorithm outperforms all of the compared algorithms.

Kacem et al.(2002 b) consider makespan, the total workload of the machines and the maximum workload of the machines objectives and develop a Pareto-optimality approach to solve the FJSP. Their approach is based on a fuzzy evolutionary optimization. The multi-objective solution quality is evaluated by a single fitness function that uses the lower-bound values on the optimal objective function values. Although their approach does not guarantee optimality, such an approach provides good quality solutions in a reasonable time limit.

Xing et al. (2008) propose a simulation model for the makespan, the total workload of the machines and the workload of the critical machine objectives. They use the weighted sum of the above three objective values as the objective function. The algorithm is compared with temporal decomposition (Kacem et al., 2002 a), classical genetic algorithm (Kacem et al., 2002), approach by localization and AL + CGA (Kacem et al., 2002 b), PSO + SA (Xia et al., 2005). The results obtained from the computational study have shown that the proposed approach is a feasible and effective approach for the multi-objective flexible job shop scheduling problem.

Shi-Jin et al. (2008) develop a new filtered-beam-search-based heuristic algorithm to solve the flexible job-shop scheduling problem with objectives of makespan, the total workload of the machines and the maximum workload of the machines. The objective function is the weighted sum of these three objective criteria. In their proposed algorithm, they design a modified branching scheme and use different dispatching rule-based heuristics as local and global evaluation functions. The performance of the proposed algorithm is compared with the temporal decomposition proposed by F. Chetouane (Kacem et al. 2002a classical genetic algorithm (Kacem et al., 2002), approach by localization and AL + CGA (Kacem et al., 2002 b), PSO + SA (Xia et al., 2005). The result of the proposed HFBS algorithm is better than those of temporal decomposition, Classical GA, AL and PSO + SA.

Gao et al. (2008) propose a new approach hybridizing genetic algorithm with variable neighborhood descent to exploit the global search ability of the genetic algorithm and the local search ability of the variable neighborhood descent for solving multi objective flexible job shop scheduling problem. The objectives are makespan, the total workload of the machines and the maximum workload of the machines. Makespan is given the first importance, maximal machine workload is given the secondary importance, and total workload is given the least importance in the study. The proposed algorithm is compared with AL + CGA (Kacem et al., 2002 b) PSO + SA (Xia et al., 2005 and multistage-based genetic algorithm (Zhang et al., 2005). Their computational results based on various benchmark problems reveal that their algorithm gives better results than the compared methods at majority of the instances.

Rajkumar et al. (2010) propose a greedy randomized adaptive search procedure (GRASP) algorithm to solve the multi-objective FJSP with maintenance constraints. The objectives are makespan, the total workload of the machines and the maximum workload of the machines. The weighted sum of the three objective values is taken as the combined objective function. The computational results based on four representative instances show the superior performance of the algorithm over the hybrid genetic algorithm.

Rajkumar et al. (2011) consider the flexible job shop scheduling problem under resource constraints. They construct a nonlinear program with three objective functions as makespan, maximum workload and total workload. The objective function of the nonlinear mathematical model is the weighted sum of the objective function values. They also present a GRASP algorithm for the weighted sum objective. They show that the GRASP gives better solutions than the GA with respect to the objective function value.

Vilcotab et al. (2011) propose two tabu search algorithms for the bicriteria FJSP with makespan and the maximum lateness objectives. The aim is to offer to the decision maker Pareto optimal solutions or non-dominated solutions. The first algorithm is based on the constrained approach and the second one is based on the linear combination of the two criteria. The proposed methods are not very attractive in the case of a single makespan criterion. For the two criteria, the method that is based on the linear combination of the two criteria produces higher quality solutions.

They compare their proposed algorithms with the tabu search algorithm proposed in Dauzere-Peres and Paulli (1997). The proposed methods are not very competitive in the case of a single criterion objective function. For two criteria case, the method based on the linear combination of criteria is the best.
Habib et al.(2012) propose a biogeography-based optimization (BBO) algorithm to solve multi objective FSJP with three different objective functions: makespan, critical machine work load, and total work load of machines. They adjust the

operators of the BBO algorithm by including migration and mutation. They compare the proposed algorithm with a newly developed GA, which has similar operators, the GA of Ho et al., the TS of Brandimart and the GA of Zhang et al. In all comparisons, the BBO algorithm shows a superior performance.
Chen et al. (2012) develop a scheduling algorithm for flexible job shop scheduling problem in reentrant process environment The objectives of the proposed algorithm are the minimization of multiple performance measures including total tardiness, total machine idle time, and makespan. The proposed algorithm includes machine selection module that selects and assigns operations to the machines and operation sequencing module that determines the processing sequence of operations on the machines considering the precedence relationships. To evaluate the performance of the proposed algorithm, they use a real weapon production factory as a case study. Simulation results demonstrate the combination of machine selection module using grouping genetic algorithm and operation sequencing module using genetic algorithm outperforms the current method used in the company.

In this study, we consider an FJSP so as to minimize the total weighted completion time. The most closely related study to ours is due to Lee et al.(2010) that considers the same environment, but total completion time criterion.

CHAPTER 3

AN APPLICATION: ROKETSAN CASE

In this study we concentrate on the flexible job shop scheduling problem of Roketsan Missiles Industries Incorporation. We introduce Roketsan, define manufacturing system, production flow and scheduling operations.

3.1. General Information About Roketsan

Roketsan was established upon a decision by Defense Industry Executive Committee at 1988 as a leader of national missile and rocket programs. To serve in the national defense sector, to contribute to the technologic infrastructure of Turkey are the missions of the company. The visions are to become the leader organization at the rocket and missile technologies from under the sea to the space and to be at the first top 50 organization in the world with its indigenous design and high technology. The company is specialized on the design and production of structural, thermal, mechanical systems; internal ballistics; guidance-control, weapon systems, aerodynamic, composite structures, propellant systems and warhead technologies.

The main customer of the company is the Turkish Armed Forces, however today it is operating beyond our borders, participating in NATO programs with its expertise and offering products for the friendly armed forces. ROKETSAN is one of the defense castles of Turkey with its 107, 122 and 302 millimeters artillery rocket systems, air defense programs, antitank missiles, precision guided munitions and airbag productions; space and satellite projects, electronic warfare support center studies.

Roketsan works on project, basis. The orders are defined by the project contracts. Aggregated capacity planning and feasibility analyses are conducted to see the attainability of the project master plan. When the contract is signed the size and time of the deliveries are specified. After receiving the orders, the company designs the operation sequence of each order, so called job, and define the processing specifications of each operation. In doing so, the capable machines for each operation together with their processing time requirements are specified. Certainly, processing time of each operation depends on both the complexity of the job and the capability of machine assigned as well as the batch size. In missile manufacturing, the process is close to being standard; jobs have predetermined operation sequences that they have to follow strictly.

3.2. The Manufacturing System

Machine capabilities are specified by the number of machine axes. A machine can have maximum 6 axes that are named as X, Y, Z, A, B and C. For the sake of completeness, below we give the function of each axis.

- *X Axis:* Relative to the part position, the tray moves through the x-axis represented by the arrow.

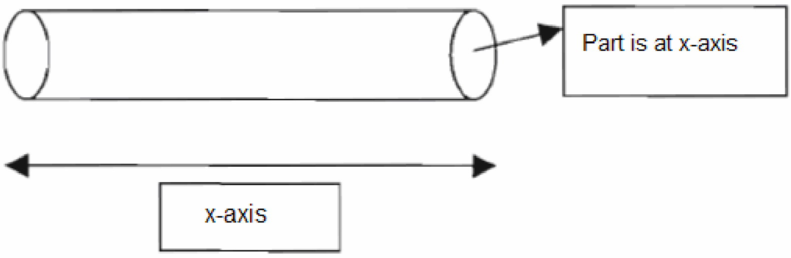

Figure 1: X-Axis Motion

- *Y Axis:* Relative to the part position, the tray moves through the y-axis represented by the arrow.

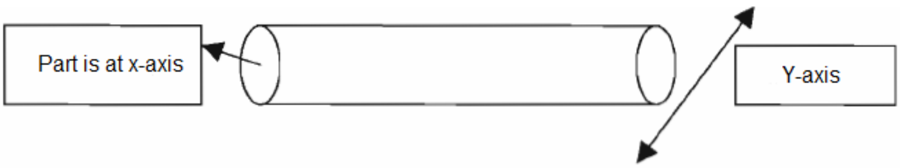

Figure 2: Y-Axis Motion

- *Z Axis:* Relative to the part position, the tray moves through the z-axis represented by the arrow.

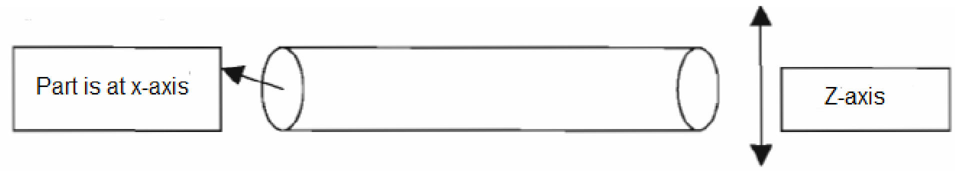

Figure 3: Z-Axis Motion

- *C Axis:* The part rotates around itself.

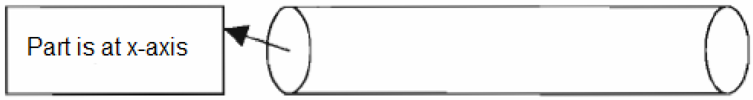

Figure 4: C-Axis Motion

- *B Axis:* The tray's angle is changeable and by this way drilling can be done to part's surface with angle different from right angle.

- *A Axis:* The spindle's angle is changeable and by this way different angled milling and drilling operations can be performed.

The manufacturing setting consists of 8 bottleneck machine types, M1 through M8. M2 has two identical machines hence there are a total of 9 machines. The characteristics of the bottleneck machines are explained below:

M1 Machine, CNC machining center: It has X, Y, Z, C and B axes and performs turning, milling and perforation and boring operations on the metal parts.

M2 Machine: There are two identical M2 with X, Z and C axes. Each can be used for turning small and precision-bored parts, processing metal and composite parts.

M3 Machine, Quality Control department: It performs precise operations at slow pace.

M4 Machine, CNC turn bench: Using X, Z and C axes, it performs various inner and outer diameter turning processes and processes metal and composite parts.
M5 Machine, CNC milling machine: Using 5-axes it performs various milling and drilling processes and processes metal and composite parts.

M6 Machine, CNC milling machine: Using X, Z and C axes, side and upper milling processes and various milling and drilling processes are performed and metal and composite parts are processed.

M7 Machine, CNC milling machine: Using X, Y, Z, C and B axes, it performs various milling processes.

M8 Machine: Using X, Y, Z and C axes various drilling especially angular drilling processes are performed and metal and composite parts are processed.
The relative locations of the machines in the manufacturing shop are given in Figure 5.

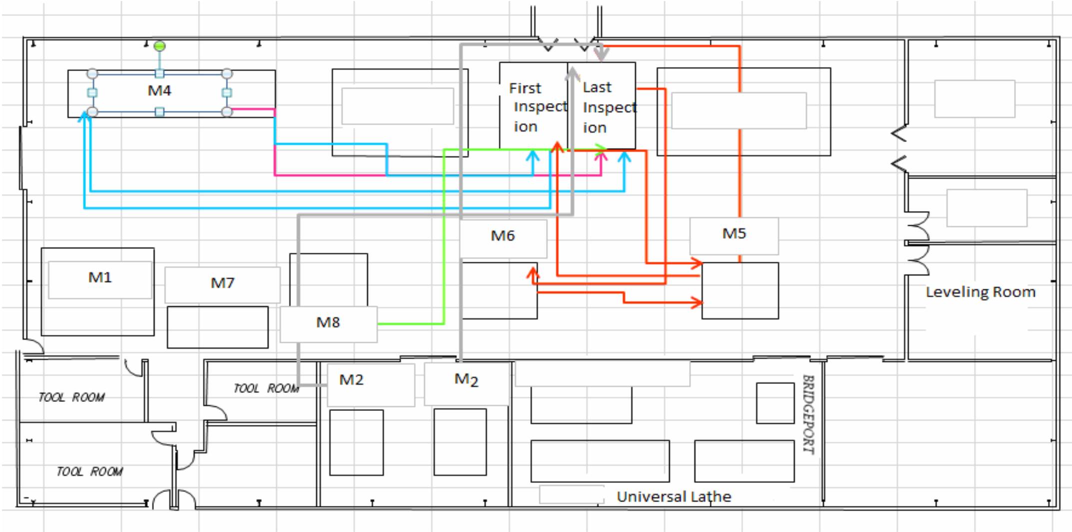

Figure 5: Workshop Layout

3.3. The Production Plan and Material Flow

The production plan is issued at the beginning of each year considering the contracts. The plan includes the number of products to be produced at each month. The monthly plans are revised to handle the deviations between the planned and realized production levels observed in the previous months. The production is done in batches.

The production department publishes the work orders with specified batch sizes. If the order, so called job, is critical, there is a first inspection point that detects whether the job is critical or not. After the batch passes the initial inspection, its first unit is processed. The processed unit is checked by the quality control department. If the unit is processed properly, the remaining units in the batch are processed at the first critical machine. If the unit is not processed properly, the machine setup is changed and one unit is processed and sent to the quality inspection department to check whether the new setup is proper. The above procedure continues until the proper output is obtained.

After all operations of job are finished, the batch is sent to quality the inspection area for final inspection after which the defectives units are scrapped. The qualified units are sent to the project depot if it is end product, else it is sent to the production depot.

The flowchart in Figure 6 depicts the above discussed production and material flow.

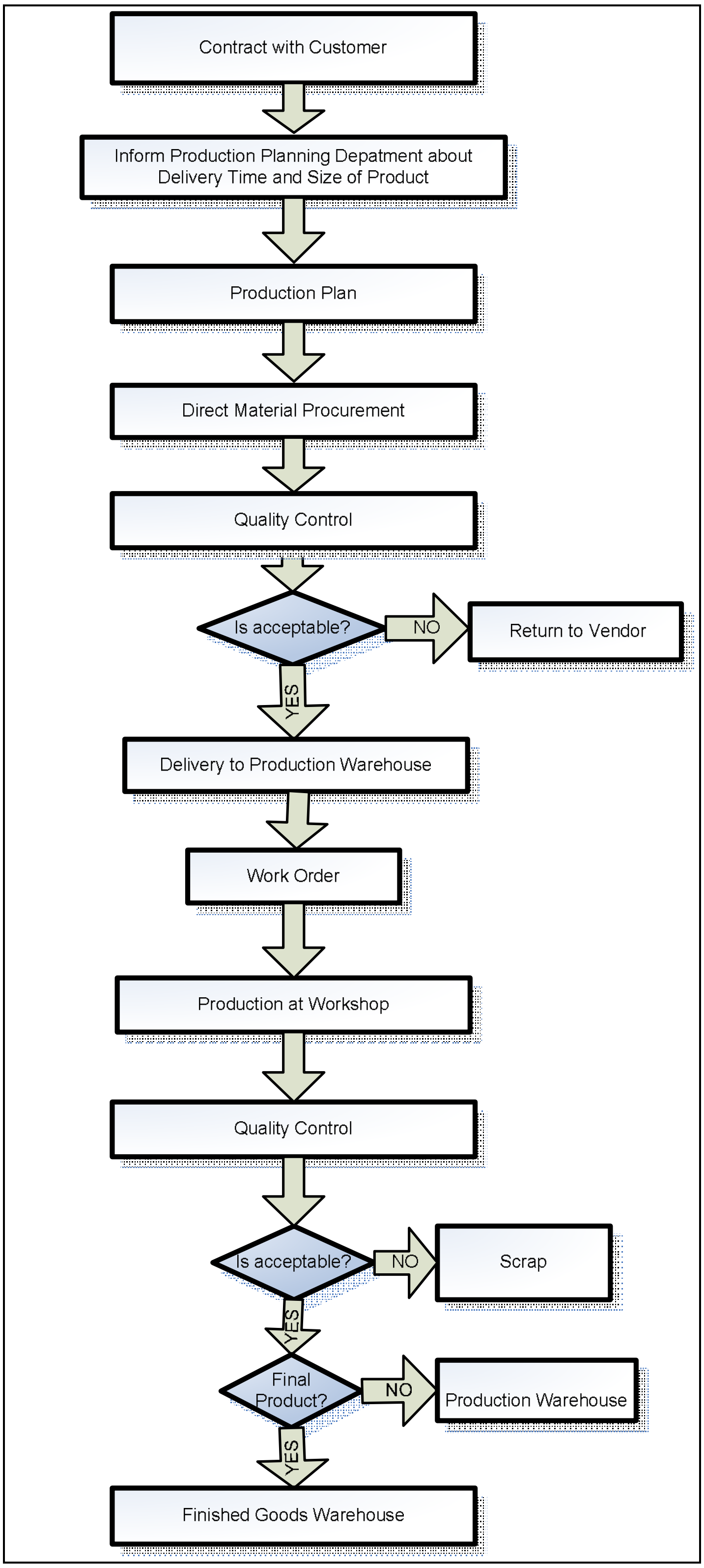

Figure 6: Production Flow Diagram

3.4. Scheduling Decisions

The scheduling problem faced by the company can be stated as allocating eight machine types with totally nine bottleneck machines to the operations and scheduling the assigned operations on each machine. Almost all projects require these nine bottleneck machines; hence their efficient planning and operation are crucial for on time deliveries and allowing new contracts.

The schedule plan is determined at the start of each week by production planning and production workshop departments and the scheduling is done in a rolling horizon basis. Each week, the parts that pent from the previous week form the machine availability constraints.

The objective is to schedule the bottleneck machines so as to finish each job as early as possible.

The following assumptions are considered in the scheduling process:

1. There is no material shortage; all resources are available when needed.

2. The system is reliable, i.e., there are no machine breakdowns and unexpected job arrivals.

3. The transfer times between the machines are not considered.

4. A machine can only process one operation at a time, and an operation is processed continuously, with no interruptions and splitting.

5. The batch sizes are set to one upon the request of the research and development department. The current scheduling policy that is determined by the chief engineer follows the below steps:

• The operations are assigned to one of their capable machines according to the minimum processing time rule and the scheduling decisions are given by the FIFO rule.
• If the deadline of a job is too close, it is started earlier than the other jobs in the cell.
• The flexible routes are defined such that the smaller indexed operation is processed earlier.

CHAPTER 4

A MIXED INTEGER LINEAR PROGRAMMING FORMULATION OF THE FLEXIBLE JOB SHOP SCHEDULING PROBLEM

In this chapter, a mixed integer linear programming (MILP) formulation of the flexible job shop scheduling problems is given. Our model finds the optimum weekly schedule on each machine. We first state the problem description and then our assumptions some of which are based on our motivating case.

The flexible job-shop scheduling problem consists of m machines and n jobs. The machine set is $M = \{M_1, M_2, \ldots , M_m\}$.Unless stated otherwise, index i denotes a machine, index j denotes a job and index h denotes an operation. Machine i has an availability time to start processing which is denoted by av_i. This parameter represents when machine i can start its first job.

Our objective is to minimize the total weighted completion time over all jobs. We assume all jobs have different penalties for late completions. w_i is the relative importance of job i. Our objective favors early completions of the jobs and gives higher priority to the more prestigious jobs. As the jobs leave the shop-floor once they are complete, the completion time of a job defines its work-in-process inventory levels.

Job j enters the shop floor at time r_j , hence r_j represents the ready time of job j. There are precedence relations between some job pairs, as some jobs cannot start before some others are completed. This may be due to the inherent assembly structure of the product.

Job j has h_j operations and follows a sequence of operation $O_{j,h}$, $h = 1, \ldots , h_j$; where $O_{j,h}$ is hth operation of job j. There are four types of operations in the system.

Type 1. An operation is performed at specified machine with a specified sequence. We let *S1* be the set of type 1 operations.

Type 2. An operation can be performed on more than one machine, hence there are alternative machines for the operation, and there is a specified sequence. We let *S2* be the set of type 2 operations.

Type 3. An operation can be performed at specified machine however at alternative sequences on its route. We let *S3* be the set of type 3 operations.

Type 4. An operation can be performed on more than one machine and at more than one sequence of its route. We let *S4* be the set of type 4 operations.

Note that Type 2 and Type 4 operations are associated with flexible nature of the job shops whereas Type 3 and Type 4 operations do follow open-shop structure.

In the absence of Type 2, 3 and 4 operations, we refer the problem as classical job shop. In the absence of Type 3 and 4 operations, we refer the problem as flexible job shop. In the presence of flexible routing structure in Type 3 and 4, our problems are open shop and flexible open shop problems, respectively. We assume the routing structure is for two specified operations of ordered h and h+k.

The batch size of job j is B_j . Batch splitting is not allowed, hence all B_j units of job j move all together. We denote $p_{i,j,h}$ as the unit processing time of $O_{j,h}$ on machine i and $S_{i,j,h}$ its setup time. Then the total processing time of job j's operation h at machine i is calculated as

$TP_{i,j,h} = S_{i,j,h} + B_j * p_{i,j,h}$, where j ϵ $M_{j,h}$, $M_{j,h}$ =set of machines that are eligible to process $O_{j,h}$

We let $p_{j,h}$ be the processing time of $O_{j,h}$ if $O_{j,h}$ can be performed by only one machine i.e. $|M_{j,h}|=1$

For the flexible routes we define the processing times as a function of machines, but not operations. We let $r_{i,j}$ be processing time of the flexible operation of job j if performed by machine i.

An instance of a flexible job shop problem is presented in Figure 7. This example describes a three jobs and three machines flexible job shop problem. The batch sizes, completion time weights and job ready times, machine available time, operations sequence, operation types, capable machines, waiting time between operations, unit processing times and setup times are shown in Figure 7.

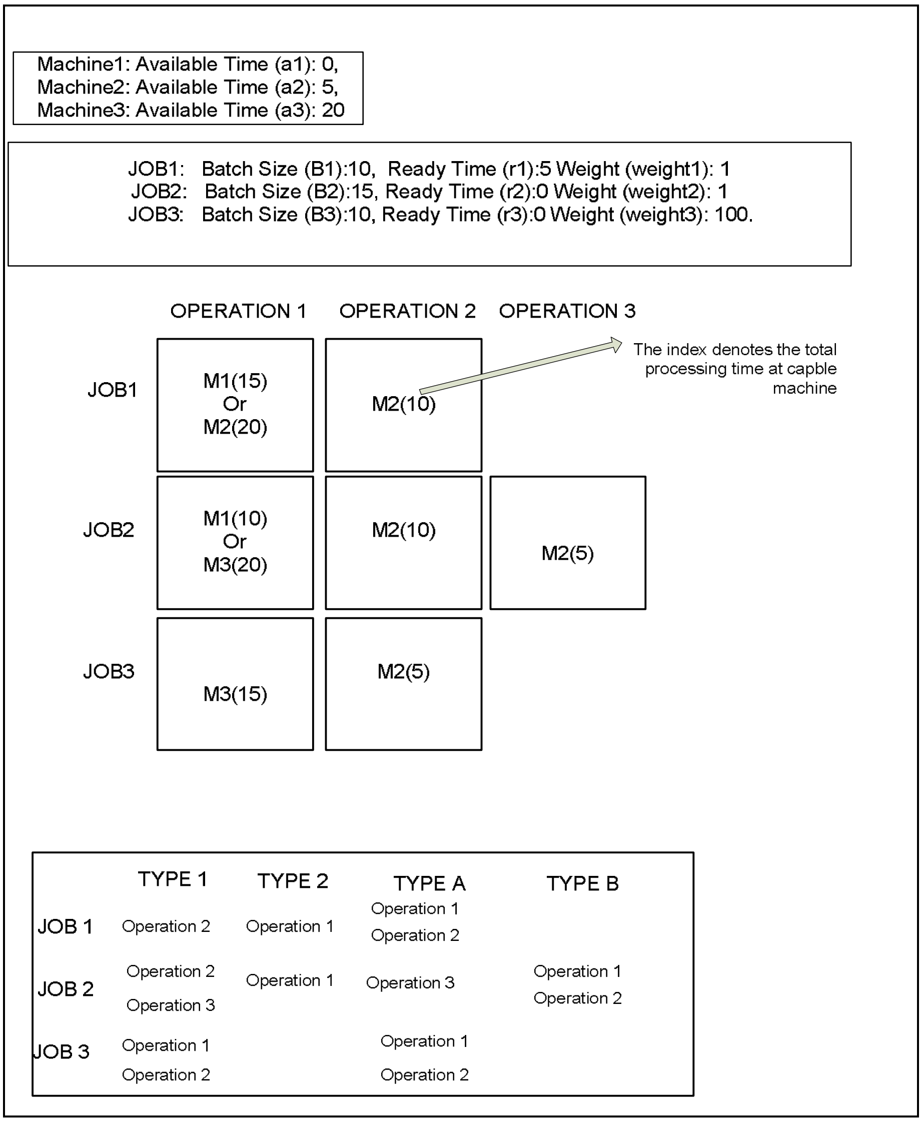

Figure 7: A Three Jobs-Three Machines Problem Instance

The MILP of the flexible job shop scheduling problem addresses the weekly schedule of the machines at the production cell. We adopt the model discussed in Fattahi et al. (2007) to our problem. Fattahi et al. (2007) aim to minimize makespan in flexible job shops. Our aim is to find an optimal weekly schedule that minimizes the total weighted flow time. Moreover our model resides the following additional features:

- flexibility in operation sequences
- ready times for the jobs
- available times for the machines

This follows the constraint set of Fattahi et al. (2007) reduces to ours when the job ready times and machine available times are all zero and all operation sequences are fixed.

We use the following additional parameter:

$a_{i,j,h}$	={1 if Oj,h can be performed on machine i and 0 otherwise}
L	= a big number

Our decision variables are

$X_{i,j,h,k}$	= {1 if $O_{j,h}$ is performed on machine i at position k and 0 otherwise where | $M_{j,h}$|>1 }
$Y_{j,h,k}$	= {1 if $O_{j,h}$ is performed on its capable machine at position k and 0 otherwise where | $M_{j,h}$|=1 }
C_j	completion time of job j
$t_{j,h}$	start time of the processing of operation $O_{j,h}$
$Tm_{i,k}$	start of working time for machine i in position k

Constraints

Time Constraints

Starting time of job j's first operation must be greater than or equal to the ready time of job j.

$$t_{j,1} \geq r_j \qquad \forall j \tag{1}$$

Start of working time for machine i must be greater than or equal to the available time of machine i.

$$Tm_{i,1} \geq av_i \qquad \forall i \tag{2}$$

Assignment Constraints

An operation can be assigned to a machine's position k if there is a capable machine to perform it.

$$y_{j,h,k} \leq \sum_i a_{i,j,h} \qquad \forall O_{j,h} \in S1 \tag{3}$$

$$x_{i,j,h,k} \le a_{i,j,h} \qquad \forall O_{j,h} \in S2, \forall i \tag{4}$$

$$x_{i,j,h,k} \le a_{i,j,h} + a_{i,j,l} \qquad \forall O_{j,h} \in S3 \cup S4 \wedge O_{j,l} \in S3 \cup S4, \forall i \tag{5}$$

Type 1 operation $O_{j,h}$ must be completely assigned to one of priorities of its capable machine.

$$\sum_k y_{j,h,k} = 1 \qquad \forall O_{j,h} \in S1 \tag{6}$$

All type 2, type3 and type4 operation $O_{j,h}$ must be assigned to one of priorities of a machine at capable machines set.

$$\sum_{i \in M_{j,h}} \sum_k x_{i,j,h,k} = 1 \qquad \forall O_{j,h} \in S2 \tag{7}$$

$$\sum_{i \in M_{j,h} \cup M_{j,l}} \sum_k x_{i,j,h,k} = 1 \qquad \forall O_{j,h} \in S3 \cup S4 \wedge O_{j,l} \in S3 \cup S4 \tag{8}$$

Type 3 and 4 operation can be performed at one of its capable machines or its correlated Type 3 or 4 operation’s one of capable machines.

$$\sum_{i \in M_{j,h} \text{ where } O_{j,h} \in S3 \cup S4} \quad \sum_{(j,h) \in S3 \cup S4} \sum_k x_{i,j,h,k} \le 1 \qquad \forall j \tag{9}$$

$$\sum_{i \in M_{j,l} \text{ where } O_{j,l} \in S3 \cup S4} \quad \sum_{(j,h) \in S3 \cup S4} \sum_k x_{i,j,h,k} \le 1 \qquad \forall j \tag{10}$$

Operation Sequence (Routing) Constraints

An operation can not start before the completion of its predecessor operations, implied by the route. This condition is handled by three constraints: first for Type 1 operations, second for Type 2 operations, third for Type 3 and Type 4 operations.

$$t_{j,h} + \sum_k y_{j,h,k} * p_{j,h} \le t_{j,h+1} \qquad \forall O_{j,h} \in S1, h \ne h_j \tag{11}$$

$$t_{j,h}+\sum_{i\in M_{j,h}}\sum_{k}x_{i,j,h,k}*p_{i,j,h}\le t_{j,h+1} \qquad \forall O_{j,h}\in S2, h\neq h_j \tag{12}$$

$$t_{j,h}+\sum_{i\in M_{j,h}\cup M_{j,l}}\sum_{k}x_{i,j,h,k}*r_{i,j}\le t_{j,h+1}$$
$$\forall O_{j,h}\in S3\cup S4\wedge O_{j,l}\in S3\cup S4, h\neq h_j \tag{13}$$

Machine Position Constraints

Machine position k represents the order of the job processed at machine i. At most one operation $O_{j,h}$ can be assigned to machine i's position k. An operation $O_{j,h}$ can be assigned to machine i's position k+1 if another operation is assigned to machine i's position k.To explain, position k+1 of machine i is open if position k is open and an operation is assigned at. The operation assigned to position (k+1) of machine i starts no earlier than the completion time of the operation assigned to the k th position of machine i.

$$\sum_{O_{j,h}\in S2}x_{i,j,h,k}+\sum_{O_{j,h}\in S1}y_{j,h,k}*a_{i,j,h}+\sum_{O_{j,h}\in S3\cup S4}x_{i,j,h,k}\le 1 \qquad \forall i,\forall k \tag{14}$$

$$\sum_{O_{j,h}\in S2}x_{i,j,h,k}+\sum_{O_{j,h}\in S1}y_{j,h,k}*a_{i,j,h}+\sum_{O_{j,h}\in S3\cup S4}x_{i,j,h,k}\ge$$
$$\sum_{O_{j,h}\in S2}x_{i,j,h,k+1}+\sum_{O_{j,h}\in S1}y_{j,h,k+1}*a_{i,j,h}+\sum_{O_{j,h}\in S3\cup S4}x_{i,j,h,k+1}$$
$$\forall i,\forall k, k\notin klast(k) \tag{15}$$

$$Tm_{i,k}+\sum_{O_{j,h}\in S1}y_{j,h,k}*p_{j,h}*a_{i,j,h}+\sum_{O_{j,h}\in S2}x_{i,j,h,k}*p_{i,j,h}+$$
$$\sum_{O_{j,h}\in S3\cup S4}x_{i,j,h,k}*r_{i,j}\le Tm_{i,k+1} \qquad \forall i,\forall k, k\notin klast(k) \tag{16}$$

Start Time of Operation's and Machine's Position Constraints

The start time of operation $O_{j,h}$ is the start of machine i's position k. that is stated in (17), (18) for Type 1 operations, in (19), (20)for Type 2 operations, (21), (22)for Type 3 and Type 4 operations.

$$Tm_{i,k} + (1 - y_{j,h,k}) * L \geq t_{j,h} \qquad \forall i \in M_{j,h}, \forall O_{j,h} \in S1, \forall k \tag{17}$$

$$Tm_{i,k} \leq (1 - y_{j,h,k}) * L + t_{j,h} \qquad \forall i \in M_{j,h}, \forall O_{j,h} \in S1, \forall k \tag{18}$$

$$Tm_{i,k} + (1 - x_{i,j,h,k}) * L \geq t_{j,h} \qquad \forall i \in M_{j,h}, \forall O_{j,h} \in S2, \forall k \tag{19}$$

$$Tm_{i,k} \leq (1 - x_{i,j,h,k}) * L + t_{j,h} \qquad \forall i \in M_{j,h}, \forall O_{j,h} \in S2, \forall k \tag{20}$$

$$Tm_{i,k} + (1 - x_{i,j,h,k}) * L \geq t_{j,h}$$
$$\forall i \in M_{j,h} \cup M_{j,l}, \forall O_{j,h} \in S3 \cup s4 \wedge O_{j,l} \in S3 \cup s4, \forall k \tag{21}$$

$$Tm_{i,k} \leq (1 - x_{i,j,h,k}) * L + t_{j,h}$$
$$\forall i \in M_{j,h} \cup M_{j,l}, \forall O_{j,h} \in S3 \cup s4 \wedge O_{j,l} \in S3 \cup s4, \forall k \tag{22}$$

Completion Time Constraints

The completion time of each job j equals to the finish time of processing of last operation.

$$C_j \geq t_{j,h} + \sum_k p_{j,h} * y_{j,h,k} \qquad \forall O_{j,h} \in S1, h = h_j \tag{23}$$

$$C_j \geq t_{j,h} + \sum_{i \in M_{j,h}} \sum_k p_{i,j,h} * x_{i,j,h,k} \qquad \forall O_{j,h} \in S2, h = h_j \tag{24}$$

$$C_j \geq t_{j,h} + \sum_{i \in M_{j,h} \cup M_{j,l}} \sum_k r_{i,j} * x_{i,j,h,k}$$
$$\forall O_{j,h} \in S3 \cup S4, h = h_j \tag{25}$$

Binary Variables and Set Constraints

$y_{j,h,k}$ and $x_{i,j,h,k}$ are binary variables:

$$y_{j,h,k} \in \{0,1\} \qquad \forall O_{j,h} \in S1, \forall k \qquad (26)$$

$$x_{i,j,h,k} \in \{0,1\} \qquad \forall O_{j,h} \in S2 \cup S3 \cup S4, \forall k \qquad (27)$$

Temporal variables should be positive.

$$C_j \geq 0 \qquad \forall j \qquad (28)$$

$$t_{j,h} \geq 0 \qquad \forall j,h \qquad (29)$$

$$Tm_{i,k} \geq 0 \qquad \forall j,k \qquad (30)$$

Objective Function

The objective function of the model is to minimize the total weighted completion times of all jobs (31).

$$minimize \ \sum_j w_j * C_j \qquad (31)$$

where w_j is the relative importance of job j

CHAPTER 5

SOLUTION APPROACHES

In this chapter, we present solution approaches for the flexible job shop scheduling problem that aims to minimize total weighted completion time. Recall that our problem is strongly NP-hard. So a small-sized problem can be solved by an optimization model and to solve medium and large-sized problems, heuristic approaches are needed.

For the FJSP there are basically two types of heuristic approaches: hierarchical approaches and integrated approaches. Hierarchical approaches are based on the idea of decomposing the original problem in order to reduce its complexity by separating the machine assignment and operation sequencing decisions. On the other hand, in integrated approaches, assignment and sequencing are not differentiated, hence are given simultaneously.

Our heuristic procedures make allocation and sequencing decisions sequentially. In doing so, we sacrifice from solution quality, with the hope of obtaining a quick solution.

Our hierarchical heuristic approach has two-phases as construction and improvement. In the construction phase, an initial assignment of the operations to one of the capable machines and sequence of the assigned operations at each machine are done. In the improvement phase, the solution built at the construction phase is improved to get a better solution in terms of the total flow time value.

5.1. Construction Phase

In the construction phase, we use hierarchical approach to generate a feasible solution for the FJSP. The output of allocation decision is the input of sequencing decision. The construction phase has two main steps as allocation and sequencing, sequencing step takes the machine assignment decisions of the allocation step as an input.

5.1.1. Allocation Decisions

We use two approaches to assign operations to one of their capable machines as assignment model and allocation rules.

5.1.1.1. Mathematical Model

Our mathematical model, so called machine assignment (allocation) model, assigns the flexible operations to one of their capable machines while ignoring the sequencing decisions. The model aims to allocate the workload between the machines as evenly as possible. In doing so, we aim to minimize the maximum workload primarily and minimize total workload secondarily. Hence we seek for a minimum total machine workload allocation among the solutions that give the smallest maximum workload.

The variables that explain our decisions are

$X_{i,j,h}$	= {1 if $O_{j,h}$ is assigned to machine i and 0 otherwise where | $M_{j,h}$|>1 }
$Y_{j,h}$	= {1 if $O_{j,h}$ is assigned to its capable machine and 0 otherwise where | $M_{j,h}$|=1 }
Z	= maximum of all machine workloads
TT_i	= workload of machine i

There are two constraint sets: assignment constraints and machine workload relations.

Assignment Constraints

Type 1 operation $O_{j,h}$ must be completely assigned to its capable machine.

$$y_{j,h} = 1 \qquad \forall O_{j,h} \in S1 \qquad (1)$$

All type 2, type3 and type 4 operations $O_{j,h}$ must be assigned to one of its capable machines.

$$\sum_{i \in M_{j,h}} x_{i,j,h} = 1 \qquad \forall O_{j,h} \in S2 \qquad (2)$$

$$\sum_{i \in M_{j,h} \cup M_{j,l}} x_{i,j,h} = 1 \qquad \forall O_{j,h} \in S3 \cup S4 \wedge O_{j,l} \in S3 \cup S4 \qquad (3)$$

Type 3 and 4 operation can be performed at one of its capable machines or its correlated Type 3 or 4 operation's one of capable machines.

$$\sum_{i \in M_{j,h} \text{ where } O_{j,h} \in S3 \cup S4} \; \sum_{O_{j,h} \in S3 \cup S4} x_{i,j,h} \leq 1 \qquad \forall j \qquad (4)$$

$$\sum_{i \in M_{j,l} \text{ where } O_{j,l} \in S3 \cup S4} \; \sum_{O_{j,h} \in S3 \cup S4} x_{i,j,h} \leq 1 \qquad \forall j \qquad (5)$$

Maximum Machine Workload Relations

The workload of a machine is the sum of the processing times of the operations assigned to that machine plus its initial available time which reflects its workload pending from previous week.

The following expression defines the workload of a particular machine i.

$$TT_i = av_i + \sum_{O_{j,h} \in S2} p_{i,j,h} * x_{i,j,h} + \sum_{O_{j,h} \in S3 \cup S4} r_{i,j} * x_{i,j,h} + \sum_{O_{j,h} \in S1} p_{j,h} * y_{j,h} \qquad \forall i \qquad (6)$$

Z gives the maximum of the machine workloads.

$$Z \geq TT_i \qquad \forall i \qquad (7)$$

Objective Function

The objective of the assignment model is to minimize

$$Z + \varepsilon * \sum_i TT_i \qquad (8)$$

where ε is a very small positive constant.

The primary objective Z aims to balance the workload by minimizing the maximum machine workload. Note that among the solutions having the smallest maximum machine workload we select the one having the total workload.

Our machine assignment is strongly NP-hard as it reduces to the well known NP-hard single stage parallel machines makespan problem (Blazewicz et al., 1994).

Owing to this complexity, we set a termination limit of 20 minutes that can be accepted by many practical applications.

5.1.1.2. SPT and Improved SPT Allocation Rule

We use the shortest processing time rule to assign the flexible operation. In doing so, we assign the flexible operations to their smallest processing time machines. This selection minimizes the total processing time. The disadvantage is that if a machine is favored by many operations due to its high speed then the resulting solution will have poor workload balance among all machines.

Recognizing this disadvantage, we improve the allocations by shifting the operations from its current machine to one of the alternate machines. In doing so, we pay attention that the load of the alternate machine after the shift is smaller than that of load of the current machine. Such a shift would balance the load of two machines at an expense of increases in the total processing time. We start from the first machine and make a shift whenever we observe that an operation when shifted to an alternate machine would help to a reduction in the machine load balance. Below is the stepwise description of our improved SPT heuristic procedure.

We use the following notation to describe our procedure

m = number of machines that reside flexible operations
$TT(i) =$ load of machine i
$p(i,j) =$ processing time of operation j on machine i
$FO(i) =$ set of flexible operations on machine i
$FF(j) =$ set of alternate machines for flexible operation j

Procedure: Improved SPT Allocation Rule

Step 0. Assign all operations to their minimum processing time machines, i.e., use SPT rule to assign operations to their flexible machines. i = 0 *Step 1.* Let $i = i + 1$ $r = 0$ If $i = m + 1$ then stop, else go to step 2. *Step 2.* Let $j = 1$ If r is the last machine in $FF(j)$ then go to Step 1 $r = r + 1$ *Step 3.* Take the j^{th} operation from set $FO(i)$ and load it to the r^{th} machine in set $FF(j)$. If $TT(r) + p(r,j) < TT(i)$ then $TT(r) = TT(r) + p(r,j)$ $TT(i) = TT(i) - p(i,j)$

$FO(i) = FO(i)/(j)$
$FO(r) = FO(r) \cup (j)$
$FF(j) = FF(j) / i \cup (r)$

If j is the last operation in $FO(i)$ then go to Step 2

Step 4. Let *j* = *j*+1
Go to Step 3.

5.1.2. Sequencing Decisions

Two approaches, a mathematical model and priority rules, are used to sequence the operations at their assigned machines.

5.1.2.1. Mathematical Model (MILP-S)

Given the machine assignment decisions, our problem reduces to the classical job shop model. The basic decision is to assign each operation to a position on each machine. The objective function of the model is minimizing the total weighted completion times, as in the original model. The only distinction from the original model is the absence of flexible operations. In this model, we assume there are no flexible operations, hence Set S2 is empty.

The resulting JSP problem still has exponential complexity however much easier as it ignores the allocation decisions. Moreover we give a termination limit of 20 minutes for the JSP model and hence control its processing speed. If no integer solution is returned at the termination limit, we use only priority rules for sequencing.

5.1.2.2. Priority Rules

We use four priority rules to sequence operations at their assigned machines. All priority rules favor our objective of minimizing total weighted completion time by giving higher priority to the operations having higher weight and lower processing time.

Priority Rule 1: In each machine, the job having the maximum weight is listed first.

Priority Rule 2: The operations are sequenced in non increasing order of the ratio of the weight of the corresponding job/total processing time of the corresponding job.

Priority Rule 3: The operations are sequenced in non decreasing order of the total processing time of the corresponding job.

Priority Rule 4: The operations are sequenced in non decreasing order of their processing time on that machine.

Each rule defines a sequence list on each machine. The resulting schedules are formed by scheduling the first feasible operation of the list whenever a machine becomes idle. An operation is feasible if all its predecessors are already assigned.

5.2. Improvement Phase

The solutions that are returned by our priority rules are subjected to an improvement process. The improvement requires a solution representation scheme, neighborhood structure given in Figure 8.

To represent the solutions, we use the sequencing list provided by Kacem et al. (2002a), in which a string embodies scheduling operations. Each operation in the solution string is represented by a triple (i, j, a(i,j)), where i represents an operation of job j and a(i,j) indicates the machine assigned to that operation. The length of the string is equal to the total number of operations of jobs.

We take a neighborhood structure, Se_1, defined by Amiri et al. (2010). *Se_1* changes the sequence of operations while preserving the precedence order between the operations and retaining the assignment of operations on the machines. In doing so, one operation among the existing operations in the current solution is selected; and if the move is precedence feasible, the operation is substituted for the operation of its preceding operation at the machine. Otherwise, the selected operation is substituted for the operation of its succeeding operation at the machine. Furthermore, it is possible that the selected operation would not be replaced with its preceding and succeeding operations at corresponding machine.

Consider the following schedule taken from the paper of Amiri et al. (2010) and depicted in Figure 8. S1 is the current solution's string from which one cell is selected at random. Suppose the second cell including operation $O_{2,1}$ is selected and is substituted at cell one and S1' is accepted as the new string.

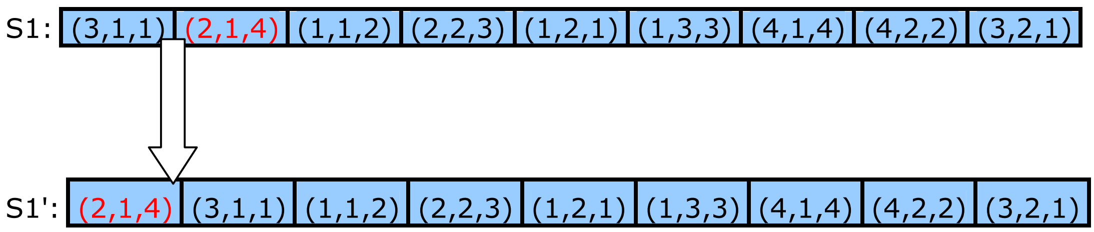

Figure 8: Example for neighborhood structure Se_1

In the neighborhood we define a move by changing the sequence of two consecutive operations. We select a move that improves the total weighted flow time, by the maximum amount.

To guide the search, we define two strategies as best improving move and best move.
Best improving move strategy selects only improving moves and stops whenever the current solution cannot be improved by realizing the moves in our neighborhood or the specified iteration limit is reached. If there is no improving

move in the neighborhood then the resulting local optimal. In such a case a better solution can be obtained only by allowing non-improving moves.

Best move recognizes this fact and continues over the best improving move by allowing the moves that deteriorate the objective function value. In order not to repeat the previously visited schedules we prevent the moves that are realized on the last three iterations on each machine.
The solution procedure for FJSP is summarized in Figure 9.

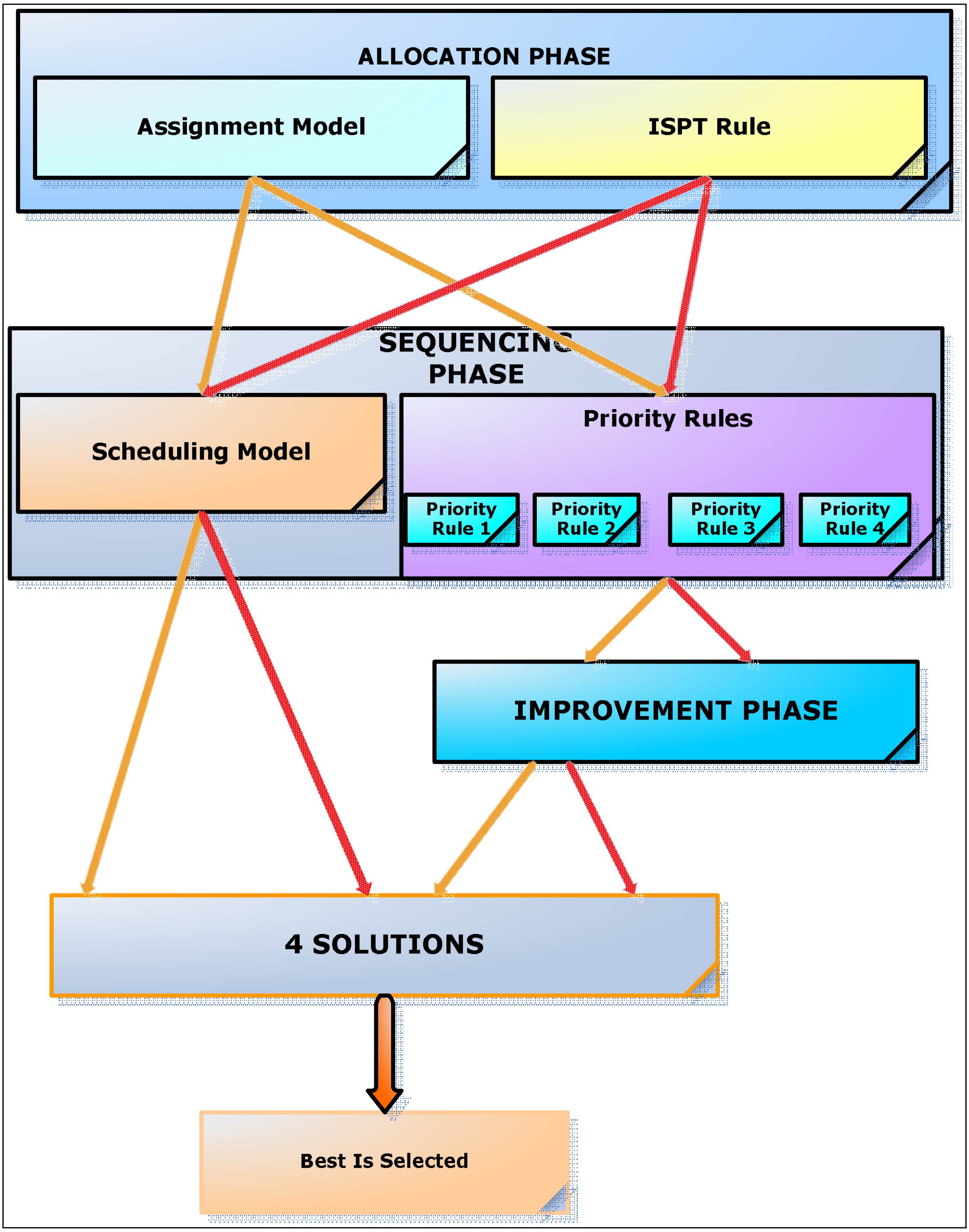

Figure 9 : Solution Procedure for FJSP Problem

CHAPTER 6

AN ILLUSTRATIVE EXAMPLE FROM ROKETSAN COMPANY

To illustrate the proposed MILP and constructive heuristic approaches, we collect 4-week data from Roketsan's production workshop. In this chapter we study week 3's case in detail. Weeks 1, 2 and 4's data and their schedules used by Roketsan production planning department, found by optimization model and heuristic methods are given in appendices.
Jobs' operations, capable machines and processing times of week 3 are reported in Table 1.

Table 1: One-Week Production System Instance

JOB	OPER.	CAPABLE M/C S	TOTAL PROCESSING TIME AT CAPABLE MACHS' (minutes)
JOB1	OPER1	MAC6	2370
	OPER2	MAC4	480
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	305
JOB2	OPER1	MAC7	960
	OPER2	MAC8	820
	OPER3	MAC6	3180
	OPER4	MAC7	1380
	OPER5	MAC6	960

Table 1 Continued

JOB2	OPER6	MAC6	600
	OPER7	MAC5,MAC19,MAC20,MAC21,MAC22	250
JOB3	OPER1	MAC10,MAC15	90
	OPER2	MAC10,MAC15	120
	OPER3	MAC10,MAC15	150
	OPER4	MAC10,MAC15	140
	OPER5	MAC5,MAC19,MAC20,MAC21,MAC22	25
JOB4	OPER1	MAC10,MAC15	270
	OPER2	MAC10,MAC15	150
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	30
JOB5	OPER1	MAC16	690
	OPER2	MAC3	2010
	OPER3	MAC16	2700
	OPER4	MAC23	780
	OPER5	MAC5,MAC19,MAC20,MAC21,MAC22	230
JOB6	OPER1	MAC10,MAC15	150
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	15
JOB7	OPER1	MAC10,MAC15	270
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	15
JOB8	OPER1	MAC9	80
	OPER2	MAC2	320
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	25
JOB9	OPER1	MAC2	1640
	OPER2	MAC3	630
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	310
JOB10	OPER1	MAC12	1030
	OPER2	MAC1	510
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	255
	OPER4	MAC12	505
	OPER5	MAC5,MAC19,MAC20,MAC21,MAC22	1010

Table 1 Continued

JOB11	OPER1	MAC10,MAC15	6030
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	255
JOB12	OPER1	MAC10,MAC15	4830
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	205
JOB13	OPER1	MAC10,MAC13,MAC15	1830
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	410
JOB14	OPER1	MAC10,MAC13,MAC15	1520
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	380
JOB15	OPER1	MAC9	435
	OPER2	MAC9	435
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	510
JOB16	OPER1	MAC10,MAC15	315
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	40
JOB17	OPER1	MAC10,MAC15	730
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	125
JOB18	OPER1	MAC14	120
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	30

In our example, last operation(s) of each job is flexible as the last operation(s) is inspection related and the company has five operators to inspect whether the part is produced correctly or not.

The machines' availability times and the jobs' ready times are given in Table 2 and Table 3, respectively. The completion times of the jobs are important criteria to define the availability times of the machines for the next week schedule. If the job does not complete at the current week, it continues to its processing at the next week. Hence the schedule has a rolling horizon aspect.

Table 2: Machine Availability Times

MACHINES	MACHINE AVAILABILITY TIME (minutes)
MAC2, MAC3, MAC6, MAC7, MAC19, MAC23	0
MAC12	300
MAC16	360
MAC20	480
MAC1, MAC5, MAC8, MAC10, MAC15, MAC21	720
MAC9, MAC13	960
MAC14, MAC22	1440

Table 3: Jobs' Ready Times

JOBS	READY TIME (min.)
JOB1,JOB5,JOB7,JOB9,JOB13,JOB14, JOB17,JOB18	0
JOB3	240
JOB6,JOB11	360
JOB16	600
JOB8	900
JOB2,JOB4,JOB15	960
JOB12	1080
JOB10	2160

Jobs 13 and 14 are relatively important; hence each is given a completion time weight 3. The weights of the other jobs are set to 1.

In section 6.1 the MILP model solution of the instance is given whereas section 6.2 reports on the heuristic solutions and 6.3 include the schedule implemented by Roketsan. Section 6.4 summarizes the completion times and calculates the objective function values of all solutions.

6.1. MILP Solutions:

We set a termination limit of 4 hours and see that the mixed integer linear programming model could not return an optimal solution. We discuss the best feasible solution returned by the model at the termination time. The job sequences are given in Table 4.

Table 4: The sequence returned by the MILP at the end of 4 hours

	Positions						
Machines	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	J10.O2						
2	J8.O2	J9.O1					
3	J5.O2	J9.O2					
4	J1.O2						
5	J16.O2	J13.O2	J10.O3	J2.O7			
6	J1.O1	J2.O3	J2.O5	J2.O6			
7	J2.O1	J2.O4					
8	J2.O2						
9	J8.O1	J15.O1	J15.O2				
10	J3.O1	J3.O2	J3.O3	J3.O4	J14.O1	J12.O1	
12	J10.O1	J10.O4					
13	J13.O1						
14	J18.O1						
15	J6.O1	J7.O1	J16.01	J4.O1	J4.O2	J17.O1	J11.O1
16	J5.O1	J5.O3					
19	J6.O2	J3.O5	J8.O3	J18.O2	J4.O3	J14.O2	J9.03
20	J7.O2	J5.O5					
21	J15.O3	J10.O5	J11.O2				
22	J17.O2	J1.O3	J12.O2				
23	J5.O4						

As can be observed from Table 4, Machines 15 and 19 are most heavily loaded by seven operations.

6.2. Heuristic Solutions:

The construction phase has two decisions as allocation and sequencing.

Allocation Decision Stage: To assign each operation to one of its capable machines we use two approaches: machine assignment model and improved shortest processing time priority rule and report the results on Table 5.

Table 5: Allocation Results

Job	Flexible Operation	Assignment Model Allocations	ISPT Allocations
JOB1	OPER3	MAC19	MAC20
JOB2	OPER7	MAC19	MAC22
JOB3	OPER1	MAC15	MAC10
	OPER2	MAC10	MAC10
	OPER3	MAC15	MAC10
	OPER4	MAC15	MAC10
	OPER5	MAC19	MAC21
JOB4	OPER1	MAC15	MAC10
	OPER2	MAC15	MAC15
	OPER3	MAC19	MAC21
JOB5	OPER5	MAC19	MAC22
JOB6	OPER1	MAC10	MAC15
	OPER2	MAC19	MAC21
JOB7	OPER1	MAC15	MAC15
	OPER2	MAC19	MAC21
JOB8	OPER3	MAC19	MAC21
JOB9	OPER3	MAC19	MAC19
JOB10	OPER3	MAC19	MAC19
	OPER5	MAC19	MAC5
JOB11	OPER1	MAC10	MAC15
	OPER2	MAC19	MAC5
JOB12	OPER1	MAC15	MAC10
	OPER2	MAC19	MAC5
JOB13	OPER1	MAC13	MAC13
	OPER2	MAC19	MAC5
JOB14	OPER1	MAC13	MAC13
	OPER2	MAC19	MAC5
JOB15	OPER3	MAC19	MAC5
JOB16	OPER1	MAC10	MAC10
	OPER2	MAC19	MAC19
JOB17	OPER1	MAC10	MAC10
	OPER2	MAC19	MAC5
JOB18	OPER2	MAC19	MAC21

Sequencing Stage: For each allocation solution, we sequence the operations in two ways: mixed integer linear model and priority rules.

Mixed Integer Linear Sequencing Model (MILP-S)

We set a termination limit of 1 hour to our MILP-S model. The sequences returned by the model that use the assignment model and improved SPT for allocation stage are reported in Tables 6 and 7, respectively.

Table 6: The Sequence Returned By MILP-S with Assignment Model

	Positions									
Mach.s	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	J10.O2									
2	J8.O2	J9.O1								
3	J5.O2	J9.O2								
4	J1.O2									
5										
6	J1.O1	J2.O3	J2.O5	J2.O6						
7	J2.O1	J2.O4								
8	J2.O2									
9	J8.O1	J15.O1	J15.O2							
10	J6.O1	J16.O1	J17.O1	J3.O2	J11.O1					
12	J10.O1	J10.O4								
13	J14.O1	J13.O1								
14	J18.O1									
15	J7.O1	J4.O1	J4.O2	J3.O1	J12.O1	J3.O3	J3.O4			
16	J5.O1	J5.O3								
19	J6.O2	J7.O2	J16.O2	J8.O3	J4.O3	J18.O2	J17.O2	J15.O3	J14.O2	J1.O3
23	J5.O4									

	Positions								
Mach.s	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
19	J9.O3	J10.O3	J13.O2	J10.O5	J3.O5	J5.O5	J11.O2	J2.O7	J5.O5

Table 7: The Sequence Returned By MILP-S with ISPT Rule

	Positions							
Mach.s	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	J10.O2							
2	J9.O1	J8.O2						
3	J5.O2	J9.O2						
4	J1.O2							
5	J14.O2	J17.O2	J15.O3	J13.O2	J10.O5	J12.O2	J11.O2	
6	J1.O1	J2.O3	J2.O5	J2.O6				

Table 7 Continued

7	J2.O1	J2.O4						
8	J2.O2							
9	J8.O1	J15.O1	J15.O2					
10	J16.O1	J3.O1	J3.O2	J3.O3	J3.O4	J4.O1	J17.O1	J12.01
12	J10.O1	J10.O4						
13	J14.O1	J13.O1						
14	J18.O1							
15	J6.O1	J7.O1	J4.O2	J11.O1				
16	J5.O1	J5.O3						
19	J16.O2	J9.O3	J10.O3					
20	J1.O3							
21	J7.O2	J8.O3	J4.O3	J6.O2	J3.O5	J18.O2		
22	J5.O5	J2.O7						
23	J5.O4							

Priority Rules

Priority Rule 1: In each machine, the job having the maximum weight is listed first. The list is 13-14-1-2-3-4-5-6-7-8-9-10-11-12-15-16-17-18 as jobs 13 and 14 have weight 3 and the weights of other jobs are one.

Priority Rule 2: The jobs are sequenced in non increasing order of the weight and total processing time ratio.

The ratios of the jobs 1 through 18 are 0.0003, 0.0001, 0.0019, 0.0022, 0.0002, 0.0061, 0.0035, 0.0024, 0.0004, 0.0003, 0.0002, 0.0002, 0.0013, 0.0016, 0.0007, 0.0028, 0.0012 and 0.0067. Hence the order is 18, 6, 7, 16, 8, 4, 3, 14, 13, 17, 15, 9, 1, 10, 12, 11, 5 and 2.

Priority Rule 3: The operations are sequenced in non-decreasing order of the total processing time of the corresponding job.

The total processing times of the jobs 1 through 18 are found as 3155, 8150, 525, 450, 6410, 165, 285, 425, 2580, 3310, 6285, 5035, 2240, 1900, 1380, 355, 855 and 150 minutes. The corresponding order is 18, 6, 7, 16, 8, 4, 3, 17, 15, 14, 13, 9, 1, 10, 12, 11, 5 and 2.

Priority Rule 4: The operations are sequenced in non decreasing order of their processing time.

We find that the best priority rule is the third for both allocation rules. Table 8 and Table 9 report the sequences returned by priority rule 3 that use allocations by the assignment model and improved ISPT rule, respectively.

Table 8: The Sequence Returned By Priority Rule 3 with Assignment Model

	Positions									
Mach.s	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	J10.O2									
2	J8.O2	J9.O1								
3	J9.O2	J5.O2								
4	J1.O2									
5										
6	J1.O1	J2.O3	J2.O5	J2.O6						
7	J2.O1	J2.O4								
8	J2.O2									
9	J8.O1	J15.O1	J15.O2							
10	J6.O1	J16.O1	J3.O2	J17.O1	J11.O1					
12	J10.O1	J10.O4								
13	J14.O1	J13.O1								
14	J18.O1									
15	J7.O1	J4.O1	J4.O2	J3.O1	J3.O3	J3.O4	J12.O1			
16	J5.O1	J5.O3								
19	J18.O2	J6.O2	J7.O2	J16.O2	J8.O3	J4.O3	J3.O5	J17.O2	J15.O3	J14.O2
23	J5.O4									

	Positions								
Mach.s	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
19	J13.O2	J9.O3	J1.O3	J10.O3	J10.O5	J12.O2	J11.O2	J5.O5	J2.O7

Table 9: The Sequence Returned By Priority Rule 3 with ISPT Rule

	Positions									
Mach.s	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	J10.O2									
2	J8.O2	J9.O1								
3	J9.O2	J5.O2								
4	J1.O2									
5	J17.O2	J15.O3	J14.O2	J13.O2	J10.O5	J12.O2	J11.O2			
6	J1.O1	J2.O3	J2.O5	J2.O6						
7	J2.O1	J2.O4								

Table 9 Continued

8	J2.O2									
9	J8.O1	J15.O1	J15.O2							
10	J16.O1	J4.O1	J3.O1	J3.O2	J3.O3	J3.O4	J17.O1	J12.O1		
12	J10.O1	J10.O4								
13	J14.O1	J13.O1								
14	J18.O1									
15	J6.O1	J7.O1	J4.O2	J11.O1						
16	J5.O1	J5.O3								
19	J16.O2	J9.O3	J10.O3							
20	J1.O3									
21	J18.O2	J6.O2	J7.O2	J8.O3	J4.O3	J3.O5				
22	J5.O5	J2.O7								
23	J5.O4									

Improvement Phase

The best of the priority rules, i.e., Priority Rule 3, with allocations by the assignment model and ISPT rule are fed into improvement phase. The sequences generated by the best improving move are reported in Tables 10 and 11. We use an iteration limit of 100 iterations for best improving move strategy. The sequences generated by best move with non-blocking strategy with 400 iterations are reported at Table 12 and 13.

Table 10: The Sequence Returned From Improvement Phase with Assignment Model (Best Improving Strategy)

	Positions									
Mach.s	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	J10.O2									
2	J9.O1									
3	J9.O2	J5.O2								
4										
5	J14.O2	J13.O2	J10.O5							
6	J2.O3	J2.O5	J2.O6							
7	J2.O4									

Table 10 Continued

8										
9	J8.O1	J15.O1	J15.O2							
10	J16.O1	J4.O1	J3.O1	J3.O3	J1.O1	J1.O2	J3.O4	J17.O1	J12.O1	J2.O1
12	J10.O1	J10.O4								
13	J14.O1	J13.O1								
14	J18.O1									
15	J6.O1	J7.O1	J11.O1							
16	J5.O1									
19										
20										
21	J8.O3	J6.O2	J4.O3	J18.O2	J3.O5					
22	J8.O2	J7.O2	J4.O2	J16.O2	J3.O2	J1.O3	J15.O3	J17.O2	J9.O3	J10.O3
23	J5.O4									

	Positions								
Mach.s	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
22	J5.O3	J5.O5	J11.O2	J12.O2	J2.O2	J2.O7			

Table 11: The Sequence Returned From Improvement Phase with ISPT Rule (Best Improving Strategy)

	Positions									
Mach.s	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	J10.O2									
2	J9.O1	J8.O2								
3	J9.O2	J5.O2								
4	J1.O2									
5	J17.O2	J15.O3	J14.O2	J13.O2	J10.O5	J12.O2	J11.O2			
6	J1.O1	J2.O3	J2.O5	J2.O6						
7	J2.O1	J2.O4								
8	J2.O2									
9	J8.O1	J15.O1	J15.O2							
10	J16.O1	J17.O1	J4.O1	J3.O1	J3.O2	J3.O3	J3.O4	J12.O1		

Table 11 Continued

12	J10.O1	J10.O4								
13	J14.O1	J13.O1								
14	J18.O1									
15	J6.O1	J7.O1	J4.O2	J11.O1						
16	J5.O1	J5.O3								
19	J16.O2	J9.O3	J10.O3							
20	J1.O3									
21	J18.O2	J6.O2	J7.O2	J8.O3	J4.O3	J3.O5				
22	J5.O5	J2.O7								
23	J5.O4									

Table 12: The Sequence Returned From Improvement Phase with Assignment Model (Best Move with Non-Blocking Strategy)

	Positions									
Mach.s	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	J10.O2									
2	J9.O1									
3	J9.O2	J5.O2								
4										
5	J14.O2	J13.O2	J10.O5							
6	J2.O3	J2.O5	J2.O6							
7	J2.O4									
8										
9	J8.O1	J15.O1	J15.O2							
10	J16.O1	J4.O1	J2.O1	J1.O1	J1.O2	J17.O1	J3.O1	J3.O3	J3.O4	J12.O1
12	J10.O1	J10.O4								
13	J14.O1	J13.O1								
14	J18.O1									
15	J6.O1	J7.O1	J11.O1							
16	J5.O1									

Table 12 Continued

19										
20										
21	J6.O2	J8.O3	J4.O3	J3.O5	J18.O2					
22	J8.O2	J7.O2	J4.O2	J16.O2	J2.O2	J1.O3	J17.O2	J3.O2	J15.O3	J9.O3
23	J5.O4									

	Positions								
Mach.s	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
22	J10.O3	J5.O3	J5.O5	J11.O2	J12.O2	J2.O7			

Table 13: The Sequence Returned From Improvement Phase with ISPT Rule (Best Move with Non-Blocking Strategy)

	Positions									
Mach.s	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	J10.O2									
2	J9.O1	J8.O2								
3	J9.O2	J5.O2								
4	J1.O2									
5	J15.O3	J14.O2	J17.O2	J13.O2	J10.O5	J12.O2	J11.O2			
6	J1.O1	J2.O3	J2.O5	J2.O6						
7	J2.O1	J2.O4								
8	J2.O2									
9	J8.O1	J15.O1	J15.O2							
10	J16.O1	J4.O1	J3.O1	J3.O2	J3.O3	J3.O4	J17.O1	J12.O1		
12	J10.O1	J10.O4								
13	J14.O1	J13.O1								
14	J18.O1									
15	J6.O1	J7.O1	J4.O2	J11.O1						
16	J5.O1	J5.O3								
19	J16.O2	J9.O3	J10.O3							
20	J1.O3									
21	J6.O2	J7.O2	J4.O3	J18.O2	J3.O5	J8.O3				
22	J5.O5	J2.O7								
23	J5.O4									

6.3. Schedule implemented in Roketsan:

The schedule constructed by the production planning department and implemented in the shop floor is given at Table 14.

Table 14: Real Schedule for Week 3

	Positions									
Mach.s	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	J10.O2									
2	J8.O2	J9.O1								
3	J5.O2	J9.O2								
4	J1.O2									
5	J1.O3	J6.O2	J10.O5	J15.O3						
6	J1.O1	J2.O3	J2.O5	J2.O6						
7	J2.O1	J2.O4								
8	J2.O2									
9	J8.O1	J15.O1	J15.O2							
10	J7.O1	J14.O1	J17.O1	J3.O1	J3.O2	J3.O3	J3.O4	J11.O1	J6.O1	J16.O1
12	J10.O1	J10.O4								
13	J13.O1									
14	J18.O1									
15										
16	J5.O1	J5.O3								
19	J2.O7	J7.O2	J11.O2	J16.O2						
20	J3.O5	J8.O3	J12.O2	J17.O2						
21	J4.O3	J9.O3	J13.O2	J18.O2						
22	J5.O5	J10.O3	J14.O2							
23	J5.O4									

	Positions								
Mach.s	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
10	J16.O1	J4.O1	J4.O2	J12.O1					

As can be observed from Table 12, Machine 10 is selected for many operations that have alternate machines and becomes the most heavily loaded machine by fourteen operations.

6.4. Evaluation of All Solutions

In this section we discuss the quality of the solutions generated by the MILP model, our heuristic procedures and the schedule implemented in Roketsan. The completion times of the schedules are reported in Table 15.

Table 15: Completion Times of Jobs Returned By the Solution Approaches

Job	MILP	MILPS		Priority Rules		Priority Rules with Best Improving Strategy		Priority Rules with Best Move with Non-Blocking Strategy		Solution Used in Roketsan
		Assignment Model	ISPT Rule	Assignment Model	ISPT Rule	Assignment Model	ISPT Rule	Assignment Model	ISPT Rule	
JOB1	3155	3235	3155	3270	3155	1950	3155	1800	3155	3155
JOB2	9110	9110	9110	9110	9110	14000	9110	7880	9110	9110
JOB3	1470	6645	2055	2035	2560	1810	2560	2475	1830	3765
JOB4	1895	1440	2015	1675	2215	1510	2215	1495	1485	10685
JOB5	6770	6875	6770	6770	7990	5290	7990	5290	7990	6770
JOB6	885	885	2030	1620	1605	1480	1605	885	885	9935
JOB7	1005	1005	1155	1605	1620	1455	1620	1455	1155	9125
JOB8	1385	1385	1985	1645	1985	1465	1985	1465	1985	3790
JOB9	4000	4000	4000	4000	2580	2950	2580	2980	2580	10995
JOB10	5470	5770	5770	5770	5730	5730	5730	5730	5730	10945
JOB11	8880	8320	8240	8320	8470	7425	8470	7425	7825	10025
JOB12	7775	6535	7570	7045	7570	7630	7570	7630	7570	15690
JOB13	3200	4720	4720	4720	4720	4720	4720	4720	4720	11405
JOB14	3130	2930	2860	2965	2860	2860	2860	2860	2860	7405
JOB15	2420	2550	3495	2585	2420	2460	2420	2670	2420	11455
JOB16	1345	1225	1075	2075	1075	1495	1075	1495	1075	10275
JOB17	2730	2040	2985	1800	1890	2640	1890	2160	2985	15815
JOB18	1590	1590	2085	1590	1590	1590	1590	1590	1590	11435

The objective function value, i.e., the total weighted completion time of the MILP solution is
1*3155+1*9110+1*1470+…+3*3200+3*3130+1*2420+1*1345+1*2730+1*1590 = 78875.

The objective function value, i.e., the total weighted completion time of the best priority rules' solution with best move with non-blocking strategy is 1*1800+1*7880+1*2475+…+3*4720+3*2860+1*2670+1*1495+1*2160+1*1590 = 77165.

The objective function value, i.e., the total weighted completion time of the schedule implemented in Roketsan is
1*3155+1*9110+1*3765+…+3*11405+3*3*7405+1*11455+1*110275+1*15815 +1*11435 = 209400.

The total weighted completion times of all schedules and the CPU times spent to find them are tabulated in Table 16.

Table 16: Total Weighted Flow Time Values of the Methods -- Week 3

Scheduling Method	Assignment Method	Total Weighted Flow Time Value (minutes)	Solution Time (CPU time)
MILP		78875	4 hours
ROKETSAN SCHEDULE		209400	-----------
MILP-S	Assignment Model	85560	1 hour
	ISPT Rule	86235	1 hour
Priority Rule 1	Assignment Model	189220	< 1 minute
	ISPT Rule	107885	< 1 minute
Priority Rule2	Assignment Model	99655	< 1 minute
	ISPT Rule	91785	< 1 minute
Priority Rule3	Assignment Model	93890	< 1 minute
	ISPT Rule	88850	< 1 minute
Priority Rule4	Assignment Model	242410	< 1 minute
	ISPT Rule	140695	< 1 minute
Improvement Phase With Best Improving Strategy	Assignment Model	83 620	< 5 minutes
	ISPT Rule	84305	< 5 minutes
Improvement Phase With Best Move with Non-Blocking Strategy	Assignment Model	77165	< 15 minutes
	ISPT Rule	82 110	< 15 minutes

Note that the best solution generated by priority rule with best move with non-blocking strategy with assignment model is superior to the best solution returned by all heuristic procedures and MILP. The best solution generated by priority rules' improvement phase with assignment model with an objective function value of 77165 is superior to the solution obtained by the MILP model which is obtained at an expense of too high computation time of 4 hours.

The objective function value of the schedule implemented in Roketsan is too high when compared to those of the heuristic solutions.

The results for other three weeks are similar and reported in Tables 17, 18 and 19 for weeks 1, 2 and 4 respectively.

Table 17: Total Weighted Flow Time Values of the Methods -- Week 1

Scheduling Method	**Assignment Method**	**Total Weighted Flow Time Value (minutes)**	**Solution Time (CPU time)**
MILP		134 026	4 hours
MILP-S	Assignment Model	208 833	1 hour
	ISPT Rule	167 263	1 hour
Priority Rule with Best Improving Strategy	Assignment Model	246 173	<5 minutes
	ISPT Rule	179 551	<5 minutes
Priority Rule with Best Move with Non-Blocking Strategy	Assignment Model	194 323	<15 minutes
	ISPT Rule	158 999	<15 minutes

Table 18: Total Weighted Flow Time Values of the Methods -- Week 2

Scheduling Method	Assignment Method	Total Weighted Flow Time Value (minutes)	Solution Time (CPU time)
MILP		No solution	4 hours
MILP-S	Assignment Model	No solution	1 hour
	ISPT Rule	No solution	1 hour
Priority Rule with Best Improving Strategy	Assignment Model	221 100	<5 minutes
	ISPT Rule	177 545	<5 minutes
Priority Rule with Best Move with Non-Blocking Strategy	Assignment Model	175 115	<15 minutes
	ISPT Rule	154 760	<15 minutes

Table 19: Total Weighted Flow Time Values of the Methods -- Week 4

Scheduling Method	Assignment Method	Total Weighted Flow Time Value (minutes)	Solution Time (CPU time)
MILP		137 570	4 hours
MILP-S	Assignment Model	No Solution	1 hour
	ISPT Rule	No Solution	1 hour
Priority Rule with Best Improving Strategy	Assignment Model	230 885	<5 minutes
	ISPT Rule	148 385	<5 minutes
Priority Rule with Best Move with Non-Blocking Strategy	Assignment Model	168 200	<15 minutes
	ISPT Rule	124 445	<15 minutes

CHAPTER 7

COMPUTATIONAL RESULTS

In this chapter we report on the performance of our mixed integer linear programming model (MILP) and heuristic approaches on randomly generated problem instances.

We generate our parameters from discrete uniform distributions whose ranges are reported in Table 20.

Table 20: Parameter Values Used at Instances

Parameter Name	Distribution
Ready Time for Each Job	Uniform between 0 and 1200
Available Time for Each Machine	Uniform between 0 and 1200
Process Time for Each Operation	Uniform between 5 and 10
Weight of Each Job	Uniform between 1 and 3
# of Alternating Operations for Each Job	0 or 2

We run the model at several problem sizes. The number of jobs, n, is increased from 3 to 7 in unit increments. For each n value, the number of operations and the number of machines and the number of alternative machines for each operation are set as in Table 21.

Table 21: The Problem Sizes Used in Our Experiments

<table>
<tr><th># of Jobs</th><th># of Operations and # of M/C s Uniform Between</th><th># of Alternative M/C s for Each Operation Uniform Between</th></tr>
<tr><td rowspan="3">3, 4, 5, 6, 7
10, 15, 20</td><td>2 and 4</td><td>1 and 2</td></tr>
<tr><td>5 and 7</td><td>1 and 3</td></tr>
<tr><td>8 and 10</td><td>1 and 3</td></tr>
</table>

7.1 Performance of the Mixed Integer Linear Model

We use our model in the following three folds:

1. To find optimal solutions (pure optimization)
Owing to the complexity of the problem, the optimization model could return optimal solutions only to the small-sized problem instances.

2. To find upper bounds by allowing 10 % relative gap from the optimal solution

3. To find lower bounds by relaxing the integrality constraints (Linear Programming Relaxations)

We solve the models with GAMS using CPLEX solver. We conduct our experiments on a computer with Intel Core i7 CPU processor 2.13 GHz.

We find that the relaxed model is solved very quickly. However this is not true for the optimal solutions and 10% gap solutions. The execution time limit for those cases is set to 1 hour as it is a tolerable solution time for real life instances.

In Section 7.1 and Section 7.2 we report on the results of MILP with no gap and 10% gap, respectively. We find that the lower bounds found through linear programming relaxations are too far from the optimal objective function values and do not report on their performances.

7.1.1 Pure Optimization (Finding Optimal Solutions)

Our aim here is to find the problem sizes at which our model returns an optimal solution in our termination limit of one hour. The CPU times and the number of unsolved instances in 1 hour are given in Table 22. The empty cells indicate that there are less than 5 optimal solutions within 10 instances.

Table 22: The CPU Times and Number of Unsolved Instances in 1 hour

OPER	2-4			5-7			8-10		
JOBS	AVG	MAX	No of unsolved instances	AVG	MAX	No of unsolved instances	AVG	MAX	No of unsolved instances
3	0.7	3.28	0	434	3600	1	657	3600	1
4	87.7	516.92	0	1903	3600	5			
5	1544	3600	4						
6	1280	3600	2						
7	2027	3600	5						

Note that the solution times increase considerably with increases in the number of jobs and number of operations. Figure 10 depicts the increases in the average CPU times with increases in the number of jobs when the number of operations is between 2 and 4.

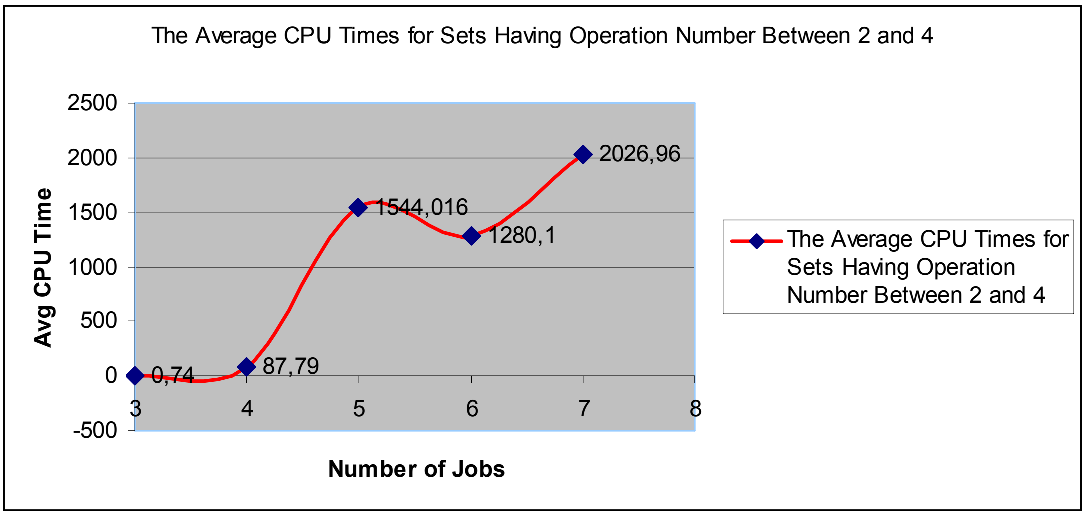

Figure 10: The Average CPU Times of Optimal Solutions - 2 and 4 operations case

7.1.2 Optimization Model with Gap 0.1 (Finding Upper Bounds)

The gap for the MILP software is defined as the difference between the cost of the best solution and the cost of its optimal linear program as a ratio of the cost of its optimal linear program. In our experiments, we use a relative gap 10% and solve the combinations where we get either an optimal solution or integer solution at our termination limit, to more than 4 optimal solutions within 10 instances in Section 7.1.

We observe the CPU times by 10% gap solutions are much smaller compared to the zero gap, i.e., optimal solutions. We give the CPU times and number of unsolved instances with 10% gap in one hour in Table 23.

Table 23: The CPU Times and Number of Unsolved Instances with 10% gap

OPER	2-4			5-7			8-10		
JOBS	AVG	MAX	No of unsolved instances	AVG	MAX	No of unsolved instances	AVG	MAX	No of unsolved instances
3	0.5	2.3	0	163.6	1448	0	158	482	0
4	5.7	40.2	0	1201	3600	3			
5	502	3600	1						
6	792	3600	2						
7	1110	3600	2						

Figure 11 plots the average CPU times of the model with gap 10% for different values of n when the number of operations is between 2 and 4.

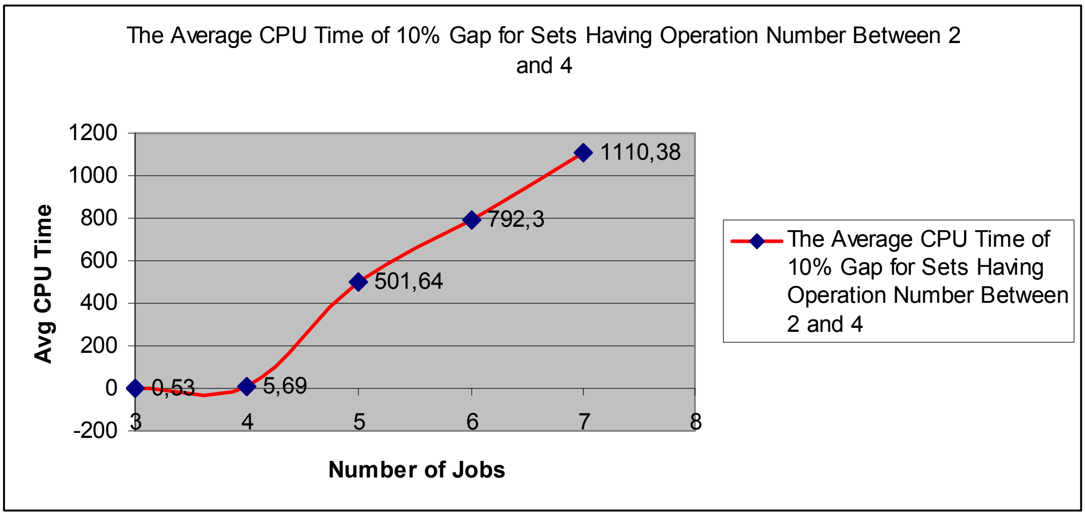

Figure 11: The Average CPU Times of 10% Gap Solutions-2 and 4 operations case

The deviations of integer solutions of model with 10% gap from the optimal solutions as a ratio of the optimal solutions and the number of compared instances are given in Table 24.

Table 24: The Deviations of 10% gap from the Optimal

	OPER 2-4			OPER 5-7			OPER 8-10		
JOBS	**AVG**	**MAX**	**No of solved instances**	**AVG**	**MAX**	**No of solved instances**	**AVG**	**MAX**	**No of solved instances**
3	2.06	9.83	10	3.71	10.48	10	5.32	10.12	10
4	1.81	6.21	10	3.82	9.01	7			
5	1.05	4.60	7						
6	3.12	10.63	8						
7	1.79	3.48	5						

As can be observed from the table the observed relative gaps are much smaller than preset gap value of 10%.

7.2 Performance of the Heuristic Procedures

We solve the sequencing and assignment model with GAMS using CPLEX solver and allocation rules and sequencing heuristic with Dev C++.

7.2.1 Preliminary Runs

Using the instances with 3 and 4 jobs, we perform a preliminary experiment, to select the allocation rules and number of iterations to be used for the best move improvement procedure, in the main runs.

We first study the performances of the machine assignments made by the allocation rules. Table 25 gives the number of instances the assignment rule gives the same machine loads with the optimal solution’s machine loads. Table 26 reports the number of machine assignments that are identical with the optimal assignments, for each of the allocation rules. Table 27 gives the associated percentages, i.e., the percentage of assignments that are identical to the optimal solutions as a ratio of the total number of machine assignments.

Table 25: Number of Instances the Assignment Rule’s Machine Loads Same with Optimal Solution’s Machine Loads

Job Number	Operation Number	Number of Instances Compared	SPT	ISPT	Assignment Model
3 JOB	OPER 8-10	9	4	1	3
	OPER 5-7	9	4	1	4
	OPER 2-4	10	8	8	8
7 JOB	OPER 2-4	5	2	2	2

Table 26: Number of Optimal Machine Assignments with Allocation Rules

		Number of Flexible Oper.s			SPT Rule			ISPT Rule			Assignment Model		
Jobs	**Number of Oper.s**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**
3	8-10	7.9	6	9	4.1	1	9	4.4	1	9	5.9	5	9
	5-7	7.7	6	9	3.2	0	8	3.3	0	8	5.2	2	9
	2-4	4.7	2	8	2.0	1	4	2.1	1	4	2.4	0	6
4	2-4	9.1	8	12	3.7	0	6	4.6	0	8	7.4	7	9

Table 27: Percentage of Optimal Machine Assignments with Allocation Rules

		Number of Flexible Oper.s			SPT Rule			ISPT Rule			Assignment Model		
Number of Jobs	**Number of Oper.s**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**	**Avg**	**Min**	**Max**
3	8-10	7.9	6	9	56.90	14.29	100.00	61.22	14.29	100.00	83.66	62.50	100.00
	5-7	7.7	6	9	40.43	0	88.89	42.02	0	88.89	69.05	22.22	100.00
	2-4	4.7	2	8	47.76	14.29	100	54.90	14.29	100.00	57.24	0.00	100.00
4	2-4	9.1	8	12	41.78	0	75	52.50	0	100.00	82.25	44.44	100.00

As can be observed from Table 26, compared to the SPT rule, the improved SPT rule results in machine assignments that are closer to the optimal machine assignments. Note that when there are 4 jobs and between 2 and 4 operations, on average 3.7 assignments made by SPT rule are identical with the optimal assignments, and 4.6 assignments made by improved SPT rule are identical with those of the optimal assignments. Using improved SPT rule increases the number of identical assignments by 10% over the SPT rule. The tables indicate the assignments made by the assignment model are very close to the optimal assignments for all tested problem sizes. Note that when there are 4 jobs and between 2 and 4 operations, on average 82.25% of the assignments made by the optimal solution and assignment model are identical. At worst case about 45% of those assignments are identical.

After analyzing the assignments made by all assignment procedures, we select the improved SPT rule and assignment model, and ignore the SPT rule. We keep the improved SPT rule, as this requires polynomial effort as opp ponential effort required by the assignment rule.

We next study the performance of the heuristic procedures nstances. Table 28 reports the percent deviation of the heuristic solution from the optimal solution as a percentage of the optimal solution for the following heuristic procedures.

1. MILPS Schedule using Assignment Model for Allocation: The assignment model solution is used to assign flexible operations to one of its candidate machines and schedule the operations at their assigned machines by using job shop scheduling model.

2. MILPS Schedule using ISPT Rule for Allocation: The improved shortest processing time rule is used to assign flexible operations to one of its candidate machines and schedule the operations at their assigned machines by using job shop scheduling model.

3. Priority Rule Schedule with best improving strategy using Assignment Model for Allocation: The assignment model solution for assigning flexible operations to one of its candidate machines is used. The best improving strategy by searching better solution than previous iteration solution and blocking the changed operations at each iteration and non-blocking all the blocked operations if the solution quality is better than the previous solution quality is used as a scheduling strategy. The heuristic ends when there is no better solution than previous iteration's solution.

4. Priority Rule Schedule with best improving strategy using ISPT Rule for Allocation: The improved shortest processing time rule is used to assign flexible operations to one of its candidate machines. The best improving strategy explained previously is used.

5. Priority Rule Schedule with best move strategy using Assignment Model for Allocation - 50 improvement steps: Assignment model solution is used to assign operations into one of their capable machines. At each iteration, the first feasible operations' interchange is taken and implemented into the previous iteration solution. In this strategy, worse solution is allowed due to the fact that better solution can possibly be generated from a worse solution. Iteration limit 50 is used.

6. Priority Rule Schedule with best move strategy using ISPT Rule for Allocation - 50 improvement steps: The improved shortest processing time rule is used to assign flexible operations to one of its candidate machines. The best move strategy explained previously with 50 iterations limit is used.

7. Priority Rule Schedule with best move strategy using Assignment Model for Allocation -- 100 improvement steps: Assignment model solution to assign operations into machines and best move strategy with iterations limit 100 are used.

8. Priority Rule Schedule with best move strategy using ISPT Rule for Allocation -- 100 improvement steps: ISPT rule to assign operations into machines and best move strategy with iterations limit 100 are used.

9. Priority Rule Schedule with combination of best move and non-blocking strategies using Assignment Model for Allocation -- 100 improvement steps: Assignment model solution is used to convert the problem from flexible job shop scheduling to job shop scheduling. In best move improvement procedures, there is a possibility that the new solutions are very close to the previously visited ones. To reduce the chance of finding such a case, we use a similar idea that is used in tabu search algorithms. We block an interchange, hence define it as tabu, if it is one of the last k realized interchanges on any one of the machines. This is kind of setting a tabure tenure of k on each machine. In preliminary runs, we try different values for k, set it to 3, 5 and 10. We find that a value of 10 returns higher quality solutions. Moreover, we remove all blocks, hence tabu moves, whenever we find a solution that replaces the current best solution. 100 iterations limit is used for the strategy.

10. Priority Rule Schedule with combination of best move and non-blocking strategies using ISPT Rule for Allocation -- 100 improvement steps: ISPT Rule to assign and priority rule schedule with best move and non-blocking strategies with 100 iterations limit is used.

Table 28 reports the average and worst case (maximum) deviations from the optimal solution. Table 29 gives the maximum and average CPU times for each of the heuristic procedures.

Table 28: Percent Deviation of the Heuristic Procedures from the Optimal Solutions

		MILP-S			
		Assignment Model		**ISPT Rule**	
		AVG	**MAX**	**AVG**	**MAX**
3 JOB	OPER 8-10	0.188	1.57	0.074	0.32
	OPER 5-7	0.05	0.184	0.05	0.184
	OPER 2-4	0.053	0.21	0.053	0.21
4JOB	OPER 2-4	0.003	0.025	0.003	0.025

Table 28 Continued

		Priority Rules(Best Improving)			
		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX
3 JOB	OPER 8-10	1.789	6.75	1.994	7.67
	OPER 5-7	1.768	8.23	1.77	8.25
	OPER 2-4	3.204	13.479	3.204	13.479
4JOB	OPER 2-4	2.92	10.451	2.92	10.451
		Priority Rules(Best Move- 50 iterations)			
		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX
3 JOB	OPER 8-10	1.811	6.69	1.811	6.69
	OPER 5-7	1.725	8.7	1.689	8.7
	OPER 2-4	2.901	13.479	2.901	13.479
4JOB	OPER 2-4	1.848	6.241	1.848	6.241
		Priority Rules (Best Move- 100 iterations)			
		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX
3 JOB	OPER 8-10	1.811	6.69	1.811	6.69
	OPER 5-7	1.689	8.7	1.689	8.7
	OPER 2-4	2.901	13.479	2.901	13.479
4JOB	OPER 2-4	1.827	6.241	1.827	6.241
		Priority Rules (Best Move with Non-blocking Strategy)			
		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX
3 JOB	OPER 8-10	1.811	6.69	1.811	6.69
	OPER 5-7	1.689	8.7	1.689	8.7
	OPER 2-4	2.901	13.479	2.901	13.479
4JOB	OPER 2-4	1.827	6.241	1.827	6.241

Table 29: The CPU times (seconds) of the Heuristic Procedures

		MILP-S			
		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX
3 JOB	OPER 8-10	6.34	23.15	5.02	10.3
	OPER 5-7	1.62	4.36	1.66	4.43
	OPER 2-4	0.47	3.11	0.46	3.65
4JOB	OPER 2-4	3.01	1.58	3.01	1.58

Table 29 Continued

		Priority Rules(Best Improving)			
		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX
3 JOB	OPER 8-10	19	28.2	16	22.5
	OPER 5-7	9.9	14.3	10.9	16.2
	OPER 2-4	4.81	8.8	4.81	8.8
4JOB	OPER 2-4	6.74	13.1	6.2	11.5
		Priority Rules(Best Move- 50 iterations)			
		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX
3 JOB	OPER 8-10	44	50.4	42.3	47
	OPER 5-7	34.76	40.3	35.1	40
	OPER 2-4	23	24.2	25.7	27
4JOB	OPER 2-4	23.7	26.3	23.9	27
		Priority Rules(Best Move- 100 iterations)			
		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX
3 JOB	OPER 8-10	70	76.4	68.28	72.7
	OPER 5-7	59.76	65.3	60.09	65.2
	OPER 2-4	44.97	46.2	47.67	49.1
4JOB	OPER 2-4	47.72	50.3	47.93	50.7
		Priority Rules (Best Move with Non-blocking Strategy- 100 iterations)			
		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX
3 JOB	OPER 8-10	70	76.4	68.28	72.7
	OPER 5-7	59.76	65.3	60.09	65.2
	OPER 2-4	44.97	46.2	47.67	49.1
4JOB	OPER 2-4	47.72	50.3	47.93	50.7

As can be observed from the tables, the quality of the solutions returned by the MILP-S scheduling rule outperforms the others. However those solutions are obtained at an expense of extremely higher CPU times. Note that priority rule schedules with best move strategy and 100 iterations and with combination of best move, non-blocking strategies with 100 iterations give higher quality solutions than other priority rule schedules. Combination of best move, non-blocking strategies with 100 iterations is selected for priority rule schedule as it non-blocks operations' changes by using the history of changes at each machine or finding a solution better than the best solution found at previous iterations. This strategy helps to escape from local optima.

7.2.2 Main Runs

To test the performance of the heuristic methods, we generate larger sized problem instances. We first evaluate the performance of the MILP-S models. In Table 30, the number of instances (out of 10), the MILP-S model gives the optimal solution is given. We give a termination limit of 1 hour for the MILP-S model and find that the optimal solutions cannot be found in this limit if there are 7 jobs and more than 7 operations and 10 jobs and more than 4 operations.

Table 30: Number of Optimal Solution Given By MILP-S Model

		Assignment Model	ISPT Model
7 JOB	OPER 2-4	8	8
	OPER 5-7	4	4
10 JOB	OPER 2-4	3	3

In our main runs, we test the performances of the MILP-S with assignment model, MILP-S with ISPT rule, the priority rules schedule with best move and non-blocking strategy. In Table 31, the number of instances each heuristic method gives the best solution is given. In Table 32, the average and maximum deviations of each heuristic from the best heuristic value are given. The empty cells indicate that MILP-S model does not give any optimal solution within the termination limit of one hour. In Table 33, the CPU times of the heuristic methods at different problem sizes are given.

Table 31: The Number of Best Solution by the Heuristic Methods

		MILP-S		Priority Rules	
		Assignment Model	ISPT Rule	Assignment Model	ISPT Rule
7 JOBS	OPER 2-4	8	8	2	2
	OPER 5-7	4	4	6	6
	OPER 8-10			7	7
10 JOBS	OPER 2-4	3	3	7	6
	OPER 5-7			5	5
	OPER 8-10			5	6
15 JOBS	OPER 2-4			10	10
	OPER 5-7			6	5
	OPER 8-10			5	6
20 JOBS	OPER 2-4			10	10
	OPER 5-7			10	10
	OPER 8-10			10	10

Table 32: Percent Deviation of Heuristic Methods from the Best Value

Deviation from Best		MILP-S				Priority Rules			
		Assignment Model		ISPT Rule		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
7 JOB	OPER 2-4	1.26	12.52	1.16	11.49	6.81	23.71	4.59	23.74
	OPER 5-7	0.00	0.00	0.00	0.00	7.76	36.05	5.39	13.99
	OPER 8-10					2.46	11.2	0.61	4.02
10 JOBS	OPER 2-4	0.00	0.00	0.00	0.00	6.22	39.55	6.17	39.55
	OPER 5-7					1.90	12.76	2.80	13.59
	OPER 8-10					0.01	0.05	1.62	4.37
15 JOBS	OPER 2-4					0.00	0.00	0.00	0.00
	OPER 5-7					0.00	0.00	0.00	0.00
	OPER 8-10					0.00	0.00	0.00	0.00
20 JOBS	OPER 2-4					0.00	0.00	0.00	0.00
	OPER 5-7					0.00	0.00	0.00	0.00
	OPER 8-10					0.00	0.00	0.00	0.00

Note from Tables 32 and 33 if the MILP-S returns a solution in one hour, it gives the best solution at all times. This is due to the fact that it finds the optimal schedule for the given machine assignment and the improvements made on the priority rules never catch this value. We also find that the solutions found by the priority rules are almost identical. This is due to the fact that the machine assignments made by the assignment model and ISPT rule are almost identical.

Table 33: The CPU times of the Heuristic Methods

		MILP-S				Priority Rules			
		Assignment Model		ISPT Rule		Assignment Model		ISPT Rule	
		AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
7 JOBS	OPER 2-4	244.6	1441.0	243.8	1449.4	81.6	102.3	80.4	103.5
	OPER 5-7	967.6	2158.0	959.8	2144.9	92.5	106.7	91.8	105.3
	OPER 8-10					96.0	108.4	97.5	109.8
10 JOBS	OPER 2-4	206.1	352.0	204.4	350.3	95.5	107.0	95.5	108.6
	OPER 5-7					99.6	108.9	98.5	105.8
	OPER 8-10					109.0	111.7	106.0	114.5
15 JOBS	OPER 2-4					97.1	108.5	96.3	109.2
	OPER 5-7					112.6	131.5	108.5	126.3
	OPER 8-10					126.7	140.8	122.3	134.6

Table 33 Continued

20 JOBS	OPER 2-4					108.8	124.0	103.7	116.5
	OPER 5-7					120.5	138.3	116.7	130.5
	OPER 8-10					160.8	182.5	149.5	167.3

Table 33 shows that the effort spent to find solutions by the MILP-S rule is considerably more compared to that of the priority rules after improvement. This is due to the fact that MILP-S requires an exponential effort as opposed to the polynomial effort required by the improvement procedures. Hence a better solution is always achieved at an expense of higher computational effort.

Figure 12 below, compares the effects of the assignment model and ISPT rule on the performance of the priority rules after the improvement.

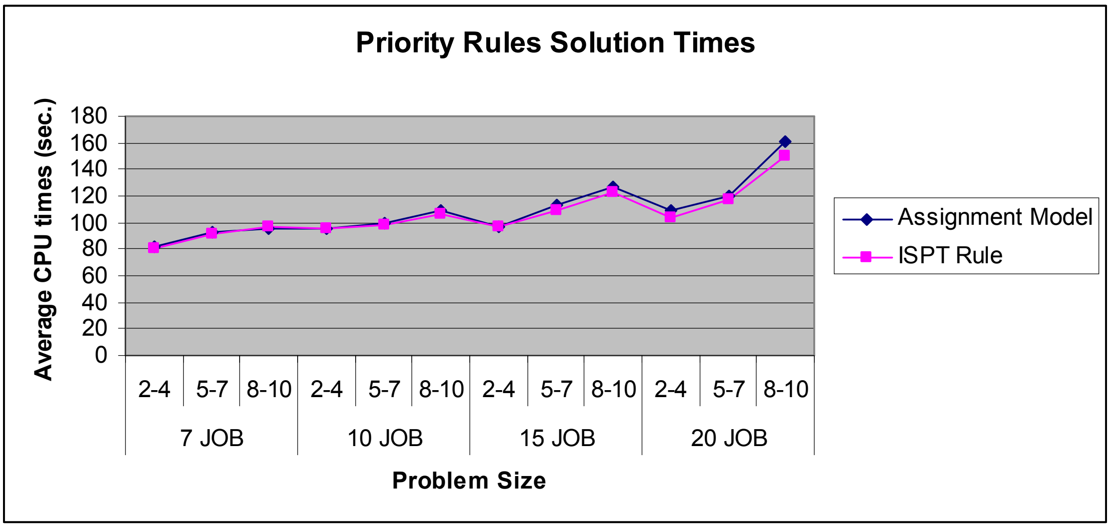

Figure 12: CPU Times of Priority Rules

As can be observed from the above figure, the CPU times increase linearly as the number of jobs and number of operations increase. Moreover any difference between the machine assignment rules on the solution times could not be observed.

To test the effect of the number of iterations on the performances of the improvement procedures, we perform runs using 200 and 400 iterations, on instances with 7 jobs and 2 to 4 operations. The percent deviations from the optimal solution and the CPU times are reported on Table 34 and Table 35, respectively.

Table 34: Deviations for the Improvement Procedures – 7 Jobs, 2-4 Operations

Priority Rules(100 iterations)				**Priority Rules(200 iterations)**				**Priority Rules(400 iterations)**			
Assignment Model		ISPT Rule		Assignment Model		ISPT Rule		Assignment Model		ISPT Rule	
AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
6.81	23.71	4.59	23.74	6.81	23.71	4.59	23.74	6.54	23.71	4.17	23.74

Table 35: CPU times for the Improvement Procedures – 7 Jobs, 2-4 Operations

Priority Rules(100 iterations)				**Priority Rules(200 iterations)**				**Priority Rules(400 iterations)**			
Assignment Model		ISPT Rule		Assignment Model		ISPT Rule		Assignment Model		ISPT Rule	
AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
81.6	102.3	80.4	103.5	110.6	136.3	109.4	134.5	139.9	168.3	138.4	164.6

Note from Table 34 that the performances slightly improve as the number of iterations increases. When the number of iterations increases from 100 to 200, there is no change in the performance. When the number of iterations increases from 200 to 400, the average deviations on the assignment model decreases from 6.81% to 6.54%. From Table 35 we see that the CPU times increase linearly with the increases in the number of iterations.

Finally, we look to the effect of the number of iterations on the performances using larger sized problem instances. Table 36 gives percent improvements made by 200 and 400 iterations on the deviations over 100 iterations. The table also includes the CPU times spent by performing 200 and 400 iterations.

Table 36: % Improvement in the Best Value over 100 iterations

			% improvement over 100 iterations		CPU Times	
Number of Jobs	Number of Operations	Iteration Number	Avg	Max	Avg	Max
7	5-7	200	0.21	0.99	182	212
		400	0.39	1.80	350	415
	8-10	200	0.49	2.59	186	220
		400	0.50	2.71	365	410
10	2-4	200	0.00	0.00	184.5	207
		400	0.00	0.00	361	398
	5-7	200	0.53	5.26	198	210
		400	0.70	5.91	394	402
	8-10	200	0.08	0.79	212	229
		400	0.18	0.96	410	445
15	2-4	200	0.92	6.35	189	214
		400	1.88	6.80	385	432
	5-7	200	0.14	1.40	220	261
		400	0.14	1.40	414	515

Note from Table 36 that the objective function values reduce by at most 1% on average, when 100 more iterations are performed. When 200 more iterations, hence 400 iterations are performed there is an additional 1% improvement. We observe that the CPU times increase linearly, they double whenever the number of iterations doubles, i.e. increases from 200 to 400.

7.3 Summary of the Computational Experiments

We first measure the performance of the MILP model in 1 hour termination limit. The termination limit is selected as 1 hour because it is tolerable for our real life application. Number of optimal solutions within 10 instances is reported for several problem sizes. The model is relaxed by relaxing the integrality constraints and it is found that the solution quality is worse for such problems. In addition the model is executed with gap value 10% and the solution quality is compared with the optimal solutions. It is found that many problems remain unsolved even with 10% gap.

We suggest a heuristic solution procedure for medium and large size problems. We study the flexible job shop scheduling problem as two sub-problems: machine assignment problem and job shop scheduling problem. For machine assignment problem, we compare three methods: SPT rule, ISPT rule and assignment model. We improve shortest processing time rule as SPT rule is likely to over-allocate some machines. In ISPT, we try to balance the workload among the machines by reallocating some operations. The assignment performances of the SPT rule. ISPT rule and assignment model are measured relative to the optimal assignments. We observe that ISPT rule dominates SPT rule and we eliminate SPT rule. The performances of the ISPT rule and assignment model are observed to be similar and both procedures are used in the main runs.

To schedule the operations on their assigned machines, two methods are used: mixed integer linear scheduling model and priority rules. The execution limit for the scheduling model is set to 1 hour. 4 priority rules used at the second method and the best of them is subjected to the improvement procedures. The improvement methods are best improving best move with 50 iterations, best move with 100 iterations and best move with non-blocking strategy and 100 iterations. Best move with non-blocking strategy and 100 iterations is found superior due to its better quality and ability to escape from local optima and hence used as an improving method in the main runs.

The performances of the heuristic methods are compared relative to the optimal solutions for small size problems. On those instances, the scheduling model dominates the priority rule solutions. However, for medium and large size problems, the scheduling model could not return optimal solutions in our termination limit of 1 hour, whereas the priority rule solutions give very quick solutions.

We observe that the number of the iterations slightly affect the performance of the improvement procedures at an expense of slightly higher solution times.

7.4 A Guide for the Selection of the Heuristics

Our experimental results have revealed that the number of jobs and the number of operations per job are the main factors that define the problem complexity and the solution methods used to solve the flexible job shop scheduling problem is selected based on these factors.

The optimization models need GAMS solver whereas the allocation and priority rules need Dev C++. If the company does not have a license to use GAMS, we propose to use the combination of allocation rule and priority rule to arrive at a feasible solution. Otherwise, in Table 37, we propose the following solution methods together with their resource (software) requirements as a function of the number of jobs and the number of operations per job.

Table 37: Solution Methods and Resource Requirements as a Function of the Problem Size

Number of Jobs	Number of Operations per Job	Solution Method	Software Requirement
less than 4	less than 8	MILP Model	GAMS
less than 4	between 8 and 10	Best of Scheduling Model and Improved Priority Rules' Solutions with Assignment Model and ISPT Rule	GAMS and Dev C++

Table 37 Continued

less than 4	more than 10	Best of Improved Priority Rules' Solution with Assignment Model and ISPT Rule	GAMS and Dev C++
between 4 and 7	between 1 and 7	Best of Scheduling Model and Improved Priority Rules' Solutions with Assignment Model and ISPT Rule	GAMS and Dev C++
between 4 and 7	more than 7	Best of Improved Priority Rules' Solution with Assignment Model and ISPT Rule	GAMS and Dev C++
between 8 and 10	between 1 and 4	Best of Scheduling Model and Improved Priority Rules' Solutions with Assignment Model and ISPT Rule	GAMS and Dev C++
between 8 and 10	more than 4	Best of Improved Priority Rules' Solution with Assignment Model and ISPT Rule	GAMS and Dev C++
more than 10	more than 1	Best of Improved Priority Rules' Solution with Assignment Model and ISPT Rule	GAMS and Dev C++

CHAPTER 8

CONCLUSIONS AND FURTHER RESEARCH DIRECTIONS

In this thesis we consider a multi-stage flexible job shop production scheduling problem existing in discrete parts manufacturing industries. We assume there are flexible operations that can be performed on one of the specified machines and flexible routes for at most two specified operations per job. Our objective is to minimize the total weighted completion times of the jobs; hence we promote early completions of the jobs.

We take our motivation from the production environment of Roketsan Missiles Industries Incorporation, operating at Turkish defense industry.

We model our flexible job shop scheduling problem (FJSP), as a mixed integer linear program (MILP). We find that the number of jobs and the number of operations are dominant factors in defining the complexity of the model. Owing to the complexity of our problem, the MILP model gives optimal solutions for small-sized problems with up to 3 jobs when the number of operations is between 8 and 10 per job and up to 5 jobs when the number of operations is between 2 and 4 per job in our plausible limit of 1 hour. Our optimization model could not return optimal solutions in 4 hours, for the real data taken from Roketsan over four weeks period, residing 18 through 23 jobs.

To solve medium and large sized problem instances, we develop heuristic algorithms. Our aim is to generate high quality approximate solutions in reasonable solution time. Our proposed heuristic has two phases as construction phase and improvement phase. In the construction phase, we use a hierarchical approach that first assigns the operations to one of its candidate machines (machine assignment step) and then finds the start and finish times of the operations at their assigned machines (scheduling step). In machine assignment step, we use a mixed integer assignment linear model that aims to balance the machine workloads and improved shortest processing time assignment rule. The workload balance problem is strongly NP-hard; however, it runs faster than the original MILP model.

In scheduling step, we use a pure job shop scheduling model and priority rules. The pure job shop scheduling model is strongly NP-hard and needs high computational time for medium-sized problems.

The solutions that are returned by the priority rules are subjected to the improvement process. The improvement phase takes the best of the priority rules to start with and uses appropriate neighborhood search structures to generate new, hopefully improved, solutions. We perform empirical analysis to fine tune the design parameters of the improvement procedures.

Our experimental results have revealed that our heuristic procedures return high quality approximate solutions. For the small sized problems, we find optimal solutions in majority of the instances. Medium sized problem instances with up to 20 jobs are solved in few minutes. Our observation is that when the scheduling step is solved by the mathematical model to optimality, the solutions have higher quality however at an expense of higher computational effort.

To the best of our knowledge our study is the first attempt to solve flexible job shop scheduling problem with total weighted completion time objective. We hope our results stimulate further research in the flexible job shop scheduling literature. Some noteworthy extensions of our work can be listed as:

- Defining appropriate batch sizes for the jobs
- Incorporating stochastic aspects of the parameters. For example, the processing times of the operations may decrease and increase as time progresses, due to the learning effect and fatigue factor, respectively.
- Handling the sequence dependent setup times
- Incorporating preventive maintenance decisions
- Defining rescheduling processes to handle unexpected breakdowns
- Using more involved procedures like shifting bottleneck procedure can be used in place of priority rules can be used to schedule operations at their assigned machines.

APPENDICES

Table 38: Week 1 Production System Instance

JOB	OPER.	CAPABLE M/Cs	TOTAL PROCESSING TIME AT ALTERNATIVE CAPABLE MACHS' (minutes)
JOB1	OPER1	MAC7	1170
	OPER2	MAC8	1000
	OPER3	MAC6	3945
	OPER4	MAC7	1710
	OPER5	MAC6	1170
	OPER6	MAC6	720
	OPER7	MAC5,MAC19,MAC20,MAC21,MAC22	460
JOB2	OPER1	MAC6	2370
	OPER2	MAC4	480
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	305
JOB3	OPER1	MAC10,MAC15	270
	OPER2	MAC10,MAC15	150
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	20
JOB4	OPER1	MAC13	103
	OPER2	MAC13	225
	OPER3	MAC13	90
	OPER4	MAC5,MAC19,MAC20,MAC21,MAC22	30
JOB5	OPER1	MAC13	850
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	35
JOB6	OPER1	MAC11	480
	OPER2	MAC11	480
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	80
JOB7	OPER1	MAC11	330
	OPER2	MAC11	630
	OPER3	MAC11	480
	OPER4	MAC5,MAC19,MAC20,MAC21,MAC22	110
JOB8	OPER1	MAC9	140
	OPER2	MAC2	440
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	65
JOB9	OPER1	MAC10,MAC15	150
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	25
JOB10	OPER1	MAC2	1640
	OPER2	MAC3	630
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	610
JOB11	OPER1	MAC10,MAC15	2430
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	405

Table 38 Continued

JOB12	OPER1	MAC10,MAC15	4530
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	760
JOB13	OPER1	MAC12	2560
	OPER2	MAC12	1510
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	505
JOB14	OPER1	MAC10,MAC15	3020
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	505
JOB15	OPER1	MAC12	1030
	OPER2	MAC1	510
	OPER3	MAC5	255
	OPER4	MAC12	505
	OPER5	MAC5,MAC19,MAC20,MAC21,MAC22	1010
JOB16	OPER1	MAC12	1030
	OPER2	MAC1	510
	OPER3	MAC5	255
	OPER4	MAC12	510
	OPER5	MAC5,MAC19,MAC20,MAC21,MAC22	505
JOB17	OPER1	MAC2	3040
	OPER2	MAC2	1090
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	810
JOB18	OPER1	MAC10,MAC15	930
	OPER2	MAC14	320
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	160
JOB19	OPER1	MAC10,MAC15	1230
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	160
JOB20	OPER1	MAC10,MAC15	4030
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	505
JOB21	OPER1	MAC12	1030
	OPER2	MAC1	1270
	OPER3	MAC5	505
	OPER4	MAC12	505
	OPER5	MAC5,MAC19,MAC20,MAC21,MAC22	255
JOB22	OPER1	MAC10,MAC15,MAC13	2370
	OPER2	MAC10,MAC15,MAC13	1590
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	200

Table 39: Week1-Machine Availability Times

MACHINE	MACHINE AVAILABILITY TIME (minutes)
MAC1, MAC3, MAC4, MAC6, MAC7, MAC8,MAC13, MAC15,MAC20, MAC21	0
MAC11, MAC19, MAC22	240
MAC2, MAC12	480
MAC5, MAC9	720
MAC10,MAC14	1440

Table 40: Week 1-Jobs' Ready Times

JOB	JOB'S READY TIME (minutes)
JOB1, JOB2, JOB5,JOB6, JOB9, JOB10, JOB11, JOB17, JOB18, JOB21	0
JOB7	240
JOB19	360
JOB12	480
JOB13, JOB20	720
JOB6, JOB16	960
JOB3, JOB4, JOB8, JOB14, JOB22	1440
JOB15	1920

Table 41: Week 1-Jobs' Weights

JOB	WEIGHT
JOB6, JOB7, JOB15, JOB16, JOB18, JOB19	3
JOB8	2
JOB1, JOB2, JOB3, JOB4, JOB5,JOB9,JOB10, JOB11, JOB12, JOB13, JOB14, JOB17, JOB20, JOB21, JOB22	1

Table 42: Week1-The sequence returned by the MILP at the end of 4 hours

	Positions							
Machines	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	J21.O2	J16.O2	J15.O2					
2	J10.O1	J8.O2	J17.O1	J17.O2				
3	J10.O2							
4	J2.O2							
5	J7.O4	J3.O3	J6.O3	J3.O3	J16.O3	J21.O3	J16.O5	J15.O4
6	J2.O1	J1.O3	J1.O5	J1.O6				
7	J1.O1	J1.O4						
8	J1.O2							
9	J8.O1							
10	J3.O1	J11.O1	J12.O1					
11	J7.O1	J7.O2	J7.O3	J6.O1	J6.O2			
12	J21.O1	J16.O1	J15.O1	J16.O4	J15.O3	J21.O4	J13.O1	
13	J5.O1	J4.O1	J4.O2	J4.O3	J22.O1	J22.O2		
14	J18.O2							
15	J9.O1	J19.O1	J3.O2	J14.O1	J20.O1	J18.O1		
19	J5.O2	J10.O3	J11.O2	J21.O5				
20	J9.O2							
21	J19.O2	J4.O4	J8.O3	J15.O5	J12.O2	J1.O7		
22	J20.O2							

	Positions				
Machines	**9**	**10**	**11**	**12**	**13**
5	J14.O2	J22.O3	J17.O3	J13.O3	J18.O3

Table 43: Week1-Completion times of jobs for MILP model

Job	Completion Time (minutes)
JOB1	10375
JOB2	3155
JOB3	1880
JOB4	1888
JOB5	885
JOB6	2720
JOB7	1790
JOB8	2625
JOB9	175
JOB10	3360

Table 43 Continued

JOB11	4545
JOB12	9430
JOB13	9665
JOB14	5385
JOB15	5850
JOB16	4585
JOB17	6690
JOB18	160
JOB19	1750
JOB20	9415
JOB21	5345
JOB22	6018

Table 44: Week1-Allocation Results

Job	Flexible Oper.	Assignment Model	ISPT Rule
JOB1	OPER7	MAC20	MAC22
JOB2	OPER3	MAC21	MAC22
JOB3	OPER1	MAC15	MAC15
	OPER2	MAC15	MAC15
	OPER3	MAC21	MAC21
JOB4	OPER4	MAC21	MAC21
JOB5	OPER2	MAC21	MAC21
JOB6	OPER3	MAC21	MAC22
JOB7	OPER4	MAC21	MAC22
JOB8	OPER3	MAC21	MAC22
JOB9	OPER1	MAC15	MAC15
	OPER2	MAC21	MAC21
JOB10	OPER3	MAC20	MAC22
JOB11	OPER1	MAC15	MAC10
	OPER2	MAC21	MAC20
JOB12	OPER1	MAC15	MAC15
	OPER2	MAC21	MAC5
JOB13	OPER3	MAC21	MAC5
JOB14	OPER1	MAC15	MAC15
	OPER2	MAC21	MAC5
JOB15	OPER5	MAC21	MAC5
JOB16	OPER5	MAC21	MAC5

Table 44 Continued

JOB17	OPER3	MAC21	MAC5
JOB18	OPER1	MAC10	MAC15
	OPER3	MAC21	MAC5
JOB19	OPER1	MAC10	MAC10
	OPER2	MAC21	MAC5
JOB20	OPER1	MAC10	MAC10
	OPER2	MAC21	MAC5
JOB21	OPER5	MAC20	MAC5
JOB22	OPER1	MAC13	MAC13
	OPER2	MAC13	MAC13
	OPER3	MAC20	MAC5

Table 45: Week1-The Sequence Returned By MILP-S with Assignment Model

	Positions							
Machines	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	J16.O2	J21.O2	J15.O2					
2	J8.O2	J10.O1	J17.O1	J17.O2				
3	J10.O2							
4	J2.O2							
5	J5.O2	J7.O4	J4.O4	J8.O3	J6.O3	J18.O3	J2.O3	J16.O3
6	J2.O1	J1.O3	J1.O5	J1.O6				
7	J1.O1	J1.O4						
8	J1.O2							
9	J8.O1							
10	J18.O1	J19.O1	J9.O1	J11.O1	J22.O1	J22.O2	J14.O1	J3.O1
11	J7.O1	J7.O2	J7.O3	J6.O1	J6.O2			
12	J21.O1	J16.O1	J15.O1	J16.O4	J21.O4	J15.O4	J13.O1	J13.O2
13	J5.O1	J4.O1	J4.O2	J4.O3				
14	J18.O2							

	Positions							
Machines	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
5	J19.O2	J9.O2	J16.O5	J21.O3	J15.O3	J10.O3	J21.O5	J15.O5
10	J3.O2	J20.O1	J12.O1					

	Positions								
Machine	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
5	J11.O2	J17.O3	J1.O7	J22.O3	J13.O3	J14.O2	J3.O3	J20.O2	J12.O2

Table 46: Week1-The Sequence Returned By MILP-S with ISPT Rule

	Positions							
Machines	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	J16.O2	J15.O2	J21.O2					
2	J8.O2	J17.O1	J17.O2	J10.O1				
3	J10.O2							
4	J2.O2							
5	J18.O3	J16.O3	J19.O2	J15.O3	J16.O5	J14.O2	J22.O3	J21.O3
6	J2.O1	J1.O3	J1.O5	J1.O6				
7	J1.O1	J1.O4						
8	J1.O2							
9	J8.O1							
10	J19.O1	J11.O1	J20.O1					
11	J7.O1	J7.O2	J7.O3	J6.O1	J6.O2			
12	J21.O1	J16.O1	J15.O1	J16.O4	J13.O1	J15.O4	J13.O2	J21.O4
13	J5.O1	J4.O1	J4.O2	J4.O3	J22.O1	J22.O2		
14	J18.O2							
15	J9.O1	J18.O1	J3.O1	J3.O2	J14.O1	J12.O1		
19								
20	J11.O2							
21	J9.O2	J5.O2	J3.O3	J4.O4				
22	J7.O4	J8.O3	J6.O3	J2.O3	J1.O7	J10.O3		

	Positions					
Machine	**9**	**10**	**11**	**12**	**13**	**14**
5	J15.O5	J17.O3	J13.O3	J21.O5	J20.O2	J12.O2

Table 47: Week1-Completion Times of Jobs Returned By MILP-S with ISPT Rule

Job Number	Completion Time (minutes)
JOB1	10375
JOB2	3155
JOB3	1880
JOB4	1910
JOB5	885
JOB6	2720
JOB7	1790
JOB8	2085
JOB9	175
JOB10	10985
JOB11	5505
JOB12	10990
JOB13	9470
JOB14	5385
JOB15	8155
JOB16	4590
JOB17	8965
JOB18	1760
JOB19	3465
JOB20	10230
JOB21	9725
JOB22	6018

Table 48: Completion Times of Jobs Returned By MILP-S with Assignment Model

Job Number	Completion Time (minutes)
JOB1	10375
JOB2	3185
JOB3	13685
JOB4	1888
JOB5	885
JOB6	2720
JOB7	1790
JOB8	2085
JOB9	3785
JOB10	5705
JOB11	7375
JOB12	22900
JOB13	11080
JOB14	13665
JOB15	6970
JOB16	4335
JOB17	8600
JOB18	2720
JOB19	3760
JOB20	18115
JOB21	5960
JOB22	10575

Table 49: Week 2 Production System Instance

Job	Oper.s	Capable M/c, S	Total Processing Time At Alternative Capable Machs' (Min.s)
JOB1	OPER1	MAC10,MAC15,MAC13	1200
	OPER2	MAC10,MAC15,MAC13	810
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	135
JOB2	OPER1	MAC10,MAC15	1830
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	460
JOB3	OPER1	MAC10,MAC13,MAC15	2730
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	155
JOB4	OPER1	MAC10,MAC15	4030
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	255
JOB5	OPER1	MAC8	660
	OPER2	MAC2	480
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	305
JOB6	OPER1	MAC11	480
	OPER2	MAC11	480

Table 49 Continued

JOB6	OPER3	MAC17	325
	OPER4	MAC17	285
	OPER5	MAC11	210
	OPER6	MAC11	390
	OPER7	MAC11	390
	OPER8	MAC11	390
	OPER9	MAC11	480
	OPER10	MAC5,MAC19,MAC20,MAC21,MAC22	100
JOB7	OPER1	MAC11	570
	OPER2	MAC11	570
	OPER3	MAC17	325
	OPER4	MAC17	285
	OPER5	MAC11	70
	OPER6	MAC5,MAC19,MAC20,MAC21,MAC22	50
	OPER7	MAC11	480
	OPER8	MAC11	70
	OPER9	MAC11	660
	OPER10	MAC5,MAC19,MAC20,MAC21,MAC22	100
JOB8	OPER1	MAC16	480
	OPER2	MAC16	240
	OPER3	MAC16	480
	OPER4	MAC16	240
	OPER5	MAC16	240
	OPER6	MAC16	240
	OPER7	MAC16	480
	OPER8	MAC16	300
	OPER9	MAC16	300
	OPER10	MAC16	300
	OPER11	MAC5,MAC19,MAC20,MAC21,MAC22	60
JOB9	OPER1	MAC9	55
	OPER2	MAC2	260
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	40
JOB10	OPER1	MAC2	3040
	OPER2	MAC3	1230
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	1210
JOB11	OPER1	MAC12	1030
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	260
JOB12	OPER1	MAC10,MAC15	1830
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	155
JOB13	OPER1	MAC12	1030
	OPER2	MAC1	510
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	255
	OPER4	MAC12	510
	OPER5	MAC5,MAC19,MAC20,MAC21,MAC22	760
JOB14	OPER1	MAC18	600
	OPER2	MAC11	1230
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	210

Table 49 Continued

JOB15	OPER1	MAC18	900
	OPER2	MAC11	2430
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	310
JOB16	OPER1	MAC2	870
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	130
JOB17	OPER1	MAC2	420
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	55
JOB18	OPER1	MAC12	365
	OPER2	MAC12	1010
	OPER3	MAC12	255
	OPER4	MAC5,MAC19,MAC20,MAC21,MAC22	1005
JOB19	OPER1	MAC9	240
	OPER2	MAC9	1350
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	170
JOB20	OPER1	MAC2	870
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	510
JOB21	OPER1	MAC2	540
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	430
JOB22	OPER1	MAC9	405
	OPER2	MAC9	405
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	470
JOB23	OPER1	MAC9	1185
	OPER2	MAC9	1185
	OPER3	MAC5,MAC19,MAC20,MAC21,MAC22	1510
JOB24	OPER1	MAC10,MAC15	520
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	55
JOB25	OPER1	MAC10,MAC15	520
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	55
JOB26	OPER1	MAC10,MAC15	310
	OPER2	MAC5,MAC19,MAC20,MAC21,MAC22	80

Table 50: Week 2- Machine Availability Times

Machines	Machine Availability Time (min.s)
MAC3, MAC9, MAC13, MAC16, MAC17, MAC18, MAC22	0
MAC8, MAC11	240
MAC2, MAC10	360
MAC21	480
MAC1	720

Table 50 Continued

MAC5	1200
MAC19	1440
MAC15	1920
MAC20	2400
MAC12	2880

Table 51: Week 2- Jobs' Ready Times

Job	Job's Ready Time (min.s)
JOB5, JOB6, JOB8, JOB10, JOB14, JOB16, JOB19, JOB20, JOB22, JOB26	0
JOB23, JOB24	240
JOB2, JOB7	480
JOB25	600
JOB12	720
JOB1, JOB9, JOB13, JOB15, JOB17	960
JOB11	1080
JOB3, JOB4, JOB21	1440
JOB18	1920

Table 52: Week2- Allocation Results

Job	Flexible Operation	Assignment Model Allocations	ISPT Rule Allocations
JOB1	OPER1	MAC10	MAC15
JOB1	OPER2	MAC10	MAC13
JOB1	OPER3	MAC22	MAC5
JOB2	OPER1	MAC10	MAC15
JOB2	OPER2	MAC22	MAC22
JOB3	OPER1	MAC10	MAC13
JOB3	OPER2	MAC22	MAC22
JOB4	OPER1	MAC10	MAC10
JOB4	OPER2	MAC22	MAC22
JOB5	OPER3	MAC22	MAC22
JOB6	OPER10	MAC22	MAC22
JOB7	OPER6	MAC22	MAC22

Table 52 Continued

JOB7	OPER10	MAC22	MAC22
JOB8	OPER11	MAC22	MAC22
JOB9	OPER3	MAC22	MAC22
JOB10	OPER3	MAC22	MAC5
JOB11	OPER2	MAC22	MAC22
JOB12	OPER1	MAC10	MAC10
JOB12	OPER2	MAC22	MAC22
JOB13	OPER3	MAC22	MAC22
JOB13	OPER5	MAC22	MAC5
JOB14	OPER3	MAC22	MAC22
JOB15	OPER3	MAC22	MAC5
JOB16	OPER2	MAC22	MAC22
JOB17	OPER2	MAC22	MAC5
JOB18	OPER4	MAC22	MAC5
JOB19	OPER3	MAC22	MAC5
JOB20	OPER2	MAC22	MAC5
JOB21	OPER2	MAC22	MAC5
JOB22	OPER3	MAC22	MAC5
JOB23	OPER3	MAC22	MAC5
JOB24	OPER1	MAC10	MAC15
JOB24	OPER2	MAC22	MAC5
JOB25	OPER1	MAC10	MAC10
JOB25	OPER2	MAC22	MAC5
JOB26	OPER1	MAC10	MAC10
JOB26	OPER2	MAC22	MAC5

Mixed Integer Linear Sequencing Model (MILP-S)

1. *Assignment Model Solution*

The assignment solution generated by assignment model solution is entered into MILP-S. The model terminates at its termination limit of 1 hour. Resource limit is exceeded and no integer solution is found.

2. *ISPT Rule Solution*

The assignment solution of improved shortest processing time rule is sequenced by scheduling linear programming model. The model terminates at its termination limit of 1 hour. Resource limit is exceeded and no integer solution is found.

Table 53: Week 4 Production System Instance

Job	Oper.	Capable Machines	Total Processing Time at Alt. Capable Mach.s
JOB1	OPER1	MAC10,MAC15	270
	OPER2	MAC10,MAC15	150
	OPER3	MAC5,MAC19,MAC20, MAC21,MAC22	20
JOB2	OPER1	MAC7	1170
	OPER2	MAC8	1000
	OPER3	MAC6	3945
	OPER4	MAC7	1710
	OPER5	MAC6	1170
	OPER6	MAC6	720
	OPER7	MAC5,MAC19,MAC20, MAC21,MAC22	310
JOB3	OPER1	MAC16	480
	OPER2	MAC16	240
	OPER3	MAC16	480
	OPER4	MAC16	240
	OPER5	MAC16	240
JOB3	OPER6	MAC16	240
	OPER7	MAC16	480
	OPER8	MAC16	300
	OPER9	MAC16	300
	OPER10	MAC16	300
	OPER11	MAC5,MAC19,MAC20, MAC21,MAC22	20
JOB4	OPER1	MAC10,MAC15	90
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	10
JOB5	OPER1	MAC9	55
	OPER2	MAC9,MAC12	420
	OPER3	MAC18	350
	OPER4	MAC18	500
	OPER5	MAC9	170
	OPER6	MAC5,MAC19,MAC20, MAC21,MAC22	110
JOB6	OPER1	MAC10,MAC15	3630
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	80
JOB7	OPER1	MAC12	1230
	OPER2	MAC1	610
	OPER3	MAC5,MAC19,MAC20, MAC21,MAC22	305
	OPER4	MAC12	605
	OPER5	MAC5,MAC19,MAC20, MAC21,MAC22	910

Table 53 Continued

JOB8	OPER1	MAC12	1030
	OPER2	MAC1	510
	OPER3	MAC5,MAC19,MAC20, MAC21,MAC22	255
	OPER4	MAC12	510
	OPER5	MAC5,MAC19,MAC20, MAC21,MAC22	505
JOB9	OPER1	MAC10,MAC15	1830
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	155
JOB10	OPER1	MAC10,MAC15	2430
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	205
JOB11	OPER1	MAC10,MAC15	4830
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	810
JOB12	OPER1	MAC10,MAC15	1230
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	55
JOB13	OPER1	MAC12	1030
	OPER2	MAC1	1270
	OPER3	MAC5,MAC19,MAC20, MAC21,MAC22	505
	OPER4	MAC12	505
	OPER5	MAC5,MAC19,MAC20, MAC21,MAC22	255
JOB14	OPER1	MAC1	330
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	165
JOB15	OPER1	MAC10,MAC15,MAC13	2730
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	155
JOB16	OPER1	MAC10,MAC15,MAC13	930
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	105
JOB17	OPER1	MAC18	425
	OPER2	MAC18	135
	OPER3	MAC9	405
	OPER4	MAC9	405
	OPER5	MAC5,MAC19,MAC20, MAC21,MAC22	260
JOB18	OPER1	MAC10,MAC15,MAC13	830
	OPER2	MAC10,MAC15,MAC13	830
	OPER3	MAC5,MAC19,MAC20, MAC21,MAC22	205
JOB19	OPER1	MAC10,MAC15,MAC13	1030
	OPER2	MAC10,MAC15,MAC13	1030
	OPER3	MAC5,MAC19,MAC20, MAC21,MAC22	255

Table 53 Continued

JOB20	OPER1	MAC10,MAC15,MAC13	830
	OPER2	MAC10,MAC15,MAC13	830
	OPER3	MAC5,MAC19,MAC20, MAC21,MAC22	205
JOB21	OPER1	MAC2	1620
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	255
JOB22	OPER1	MAC2	870
	OPER2	MAC5,MAC19,MAC20, MAC21,MAC22	510

Table 54: Week 4- Machine Availability Times

Machine	Machine Availability Time (minutes)
MAC1, MAC8, MAC9, MAC12, MAC13, MAC18, MAC22	0
MAC10, MAC16	480
MAC2	600
MAC5, MAC7, MAC19	960
MAC6	1440
MAC15	1920
MAC20, MAC21	2400

Table 55: Week 4- Jobs' Ready Times

Job	Job's Ready Time (minutes)
JOB2, JOB3, JOB11, JOB14, JOB17, JOB18, JOB20, JOB21, JOB22	0
JOB13	360
JOB19	480
JOB1, JOB9, JOB12	720
JOB6, JOB8, JOB16	1200
JOB4, JOB10	1440
JOB15	1680
JOB5, JOB7	1920

Table 56: Week 4- Jobs' Weights

Job	Weight
JOB7, JOB14	3
JOB18, JOB20	2
JOB1, JOB2, JOB3, JOB4, JOB5, JOB6, JOB8, JOB9, JOB10, JOB11, JOB12, JOB13, JOB15, JOB16, JOB17, JOB19, JOB21, JOB22	1

Table 57: Week 4-The sequence returned by the MILP at the end of 4 hours

	Positions									
Machines	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
MAC1	J14.O1	J13.O2	J7.O3	J8.O2						
MAC2	J22.O1	J22.O2								
MAC5	J22.O2	J4.O2	J5.O6	J10.O2	J15.O2	J8.O3	J12.O2	J8.O5	J9.O2	J19.O3
MAC6	J2.O3	J2.O5	J2.O6							
MAC7	J2.O1	J2.O4								
MAC8	J2.O2									
MAC9	J17.O3	J17.O4	J5.O1	J5.O5						
MAC10	J11.O1	J4.O1	J12.O1	J6.O1						
MAC12	J13.O1	J7.O1	J13.O4	J5.O2	J7.O4	J8.O1	J8.O4			
MAC13	J18.O1	J18.O2	J16.O1	J15.O1	J19.O1					
MAC15	J20.O1	J20.O2	J1.O1	J1.O2	J10.O1	J9.O1	J19.O2			
MAC16	J3.O1	J3.O2	J3.O3	J3.O4	J3.O5	J3.O6	J3.O7	J3.O8	J3.O9	J3.O10
MAC18	J17.O1	J17.O2	J5.O3	J5.O4						
MAC19	J20.O3	J3.O11	J13.O5	J7.O5	J3.O11					
MAC20	J1.O3									
MAC21	J11.O2									
MAC22	J14.O2	J17.O5	J18.O3	J13.O3	J7.O2	J21.O2	J16.O2	J6.O2		

Table 58: Week 4-Completion times of jobs for MILP model

JOB	COMPLETION TIME (minutes)
JOB1	4020
JOB2	10985
JOB3	3805
JOB4	5410
JOB5	5520
JOB6	10340
JOB7	5605
JOB8	8060
JOB9	8415
JOB10	6635
JOB11	6120
JOB12	7100
JOB13	4060
JOB14	495
JOB15	6790
JOB16	3830
JOB17	1630
JOB18	1865
JOB19	9545
JOB20	3785
JOB21	3725
JOB22	1980

Table 59: Week4-Allocation Results

Job	Flexible Operation	Assignment Model Allocations	ISPT Rule Allocations
JOB1	OPER1	MAC10	MAC10
	OPER2	MAC10	MAC10
	OPER3	MAC22	MAC5
JOB2	OPER7	MAC22	MAC22
JOB3	OPER11	MAC22	MAC22
JOB4	OPER1	MAC10	MAC10
	OPER2	MAC22	MAC22
JOB5	OPER2	MAC9	MAC9
	OPER6	MAC22	MAC22
JOB6	OPER1	MAC10	MAC10
	OPER2	MAC22	MAC22
JOB7	OPER3	MAC22	MAC22
	OPER5	MAC22	MAC22
JOB8	OPER3	MAC22	MAC22
	OPER5	MAC22	MAC5

Table 59 Continued

JOB9	OPER1	MAC15	MAC10
	OPER2	MAC22	MAC5
JOB10	OPER1	MAC10	MAC15
	OPER2	MAC22	MAC5
JOB11	OPER1	MAC15	MAC15
	OPER2	MAC22	MAC5
JOB12	OPER1	MAC10	MAC10
	OPER2	MAC22	MAC22
JOB13	OPER3	MAC22	MAC5
	OPER5	MAC22	MAC5
JOB14	OPER2	MAC22	MAC5
JOB15	OPER1	MAC10	MAC13
	OPER2	MAC22	MAC5
JOB16	OPER1	MAC10	MAC13
	OPER2	MAC22	MAC5
JOB17	OPER5	MAC22	MAC5
JOB18	OPER1	MAC10	MAC13
	OPER2	MAC10	MAC13
	OPER3	MAC22	MAC5
JOB19	OPER1	MAC10	MAC13
	OPER2	MAC10	MAC13
	OPER3	MAC22	MAC5
JOB20	OPER1	MAC10	MAC13
	OPER2	MAC10	MAC10
	OPER3	MAC22	MAC5
JOB21	OPER2	MAC22	MAC5
JOB22	OPER2	MAC22	MAC5

Mixed Integer Linear Sequencing Model (MILP-S)

1. *Assignment Model Solution*

The assignment solution generated by assignment model solution is sequenced by MILP-S. The model terminates at its termination limit of 1 hour and returns no solution. Resource limit is exceeded and no integer solution is found.

2. *ISPT Rule Solution*

The assignment solution of improved shortest processing time rule is sequenced by scheduling linear programming model. The model terminates at its termination limit of 1 hour and returns no solution. Resource limit is exceeded and no integer solution is found.

REFERENCES

Akhshabi M., Akhshabi M., Khalatbari J. (2011), "Solving flexible job-shop scheduling problem using clonal selection algorithm", Indian Journal of Science and Technology, 4, 6974- 6846.

Al-Hinai N., ElMekkawy T. Y. (2011), "An efficient hybridized genetic algorithm architecture for the flexible job shop scheduling problem", Flexible Service and Manufacturing Journal, 23, 64–85.

Amiri M., Zandieh M., Yazdani M., Bagher A. (2010), "A variable neighbourhood search algorithm for the flexible job-shop scheduling problem", International Journal of Production Research, 48, 5671–5689.

Brandimarte P. (1993), "Routing and scheduling in a flexible job shop by tabu search", Annals of Operations Research, 41, 157-183.

Brandimarte P., Calderini M. (1995), "A hierarchical bicriterion approach to integrated process planning and job shop scheduling", International Journal of Production Research, 33, 161-183.

Bruker P., Schlie R. (1990), "Job-shop scheduling with multi-purpose machines", Computing, 45, 369-375.

Chen J.C., Cheng-Chun Wu C., Chen C., Chen K. (2012), "Flexible job shop scheduling with parallel machines using genetic algorithm and grouping genetic algorithm", Expert Systems with Applications, 39, 10016 – 10021.

Fattahi P., Saidi M., Jolai F. (2007), "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems", Journal of Intelligent Manufacturing, 18, 331–342.

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

Gao J. Sun L.Y., Gen M. (2008), "A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems", Computers and Operations Research, 35, 2892–2907.

Geoffrey V. G., Billaut J. C. (2011), "A tabu search algorithm for solving a multicriteria flexible job shop scheduling problem", International Journal of Production Research, 49, 6963–6980.

Habib S., Rahmati A., Zandieh M. (2012), "A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem", International Journal of Advanced Manufacturing Technology, 58, 1115–1129.

Ham M., Lee Y. L., Kim S. H. (2011), "Real-time scheduling of multi-stage flexible job shop floor", International Journal of Production Research, 49, 3715–3730.

Ida K., Oka K. (2009), "Flexible job-shop scheduling problem by genetic algorithm", Electrical Engineering in Japan, 177, 505–511.

Kacem I., Hammadi S., Borne P. (2002), "Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems", IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 32, 1–13.

Kacem I., Hammadi S., Borne P. (2002), "Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic", Mathematics and Computer Simulation, 60, 245–276.

Lee S., Moon I., Bae H., Kim J. (2012), "Flexible job-shop scheduling problems with AND/OR precedence constraints", International Journal of Production Research, 50, 1979–2001.

Lei D., Guo X. (2012), "Swarm-based neighbourhood search algorithm for fuzzy flexible job shop scheduling", International Journal of Production Research, 50, 1639–1649.

Ling Wang L., Zhou G., Xu Y., Wang S., Liu M. (2012), "An effective artificial bee colony algorithm for the flexible job-shop scheduling problem", International Journal of Advanced Manufacturing Technology, 60, 303–315.

Li J., Pan Q., Chen J. (2012), "A hybrid Pareto-based local search algorithm for multi-objective flexible job shop scheduling problems", International Journal of Production Research, 50, 1063–1078.

Liu J., MacCarthy B.L. (1996), "A global MILP model for FMS scheduling", European Journal of Operational Research, 100, 441-453.

Li J., Pan Q., Suganthan P., Chua T. (2011), "A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem", International Journal of Advanced Manufacturing Technology, 52, 683–697.

Mati Y., Lahlou C., Dauzere-Peres S. (2011), "Modelling and solving a practical flexible job-shop scheduling problem with blocking constraints", International Journal of Production Research, 49, 2169–2182.

Rajkumar M., Asokan P., Vamsikrishna V. (2010), "A GRASP algorithm for flexible job-shop scheduling with maintenance constraints", International Journal of Production Research , 48, 6821–6836.

Rajkumar M., Asokan P., Anilkumar N., Page T. (2011), "A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints", International Journal of Production Research, 49, 2409–2423.

Shi-Jin W., Bing-Hai Z., Li-Feng X. (2008), "A filtered-beam-search-based heuristic algorithm for flexible job-shop scheduling problem", International Journal of Production Research, 46, 3027–3058.

Saidi-Mehrabad M., Fattahi P. (2007), "Flexible job shop scheduling with tabu search algorithms", Intelligent Journal of Advance Manufacturing Technology, 32, 563–570.

Taneva I.T., Uozumi T., Morotome Y. (2004), "Hybrid evolutionary algorithm-based real-world flexible job shop scheduling problem: application service provider approach", Applied Soft Computing, 5, 87–100.

Urlings T., Ruiz R. (2010), "Genetic algorithms with different representation schemes for complex hybrid flexible flow line problems", International Journal of Metaheuristics, 1, 30-54.

Xing L., Chen Y., Yang K. (2009), "Multi-objective flexible job shop schedule: design and evaluation by simulation modeling", Applied Soft Computing, 9, 362–376.

Xing L., Ying-Wu Chen Y., Yang K. (2011), "Multi-population interactive coevolutionary algorithm for flexible job shop scheduling problems", Computational Optimization and Applications, 48, 139–155.