ABSTRACT

LAOBOONLUR, PREECHA. Production Scheduling in Knitted Fabric Dyeing and Finishing: A Case Approach. (Under the direction of Thom J. Hodgson)

The dyeing and finishing processes represent one of the most complicated scheduling problems existing in real production. The problem combines two difficult but challenging scheduling aspects together: a flexible job shop with sequence dependent setups. The process consists of multiple operations, which can have either single or parallel machines. Chemical and fabric pile contamination cause the sequence dependent setups. According to the business strategy of the case factory, the scheduling problem is categorized as two cases, no job priority and two-job priority classification (high and low). The scheduling objective is to minimize maximum lateness, $L_{\max}$.

The fundamental structure used for solving the dyeing and finishing scheduling problem is the Virtual Factory plus family scheduling. The Virtual Factory is a simulation based job shop scheduling system developed at North Carolina State University. The scheduling heuristic used in the Virtual Factory is developed based on family scheduling. Jobs are grouped into families and then families are scheduled. The schedule is accomplished by switching the positions and splitting the family members.

This dissertation intends to solve the real problem. Scheduling problems are generated using real problem characteristics. The experimentation indicates that with the advantages of fast computation time and heuristics modified easily, the best approach is to apply several versions of a heuristic to get the best possible solution.

PRODUCTION SCHEDULING IN KNITTED FABRIC DYEING AND FINISHING: A CASE APPROACH

by

PREECHA LAOBOONLUR

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

INDUSTRIAL ENGINEERING

Raleigh

2004

APPROVED BY:

Dr. Thom J. Hodgson
Chair of Advisory Committee

Dr. Russell E. King

Dr. Henry L. W. Nuttle

Dr. Kristin A. Thoney

Biography

Preecha Laoboonlur was born on February 20, 1975 in Bangkok, Thailand. He graduated from Chulalongkorn University, Bangkok, Thailand with a Bachelor of Industrial Engineering in May, 1998. He received a Master of Science in Engineering Business Management in June 2000, from the joint graduate program between the University of Warwick and Chulalongkorn University, Thailand. He worked for three years in the Pacific Dyeing & Finishing Co., Ltd., a textile company in Samutsakorn, Thailand, both while he was studying and after he finished his Masters' degree.

He then entered the North Carolina State University to pursue his Ph.D. in Industrial Engineering, in January of 2001. While he was working on his Ph.D. degree, he received a Master of Textiles in 2003, under the direction of Dr. Helmut H. Hergeth. Preecha completed his Ph.D. work in 2004, under the direction of Dr. Thom J. Hodgson.

Acknowledgement

I wish to express my most sincere gratitude and appreciation to my advisor. Prof. Thom J. Hodgson, for his guidance, support and the insights he had, which made this dissertation possible. I admire the persistent attitudes, personality, and his sense of humor, which is one of the most valuable lessons for my life.

I am grateful to Dr. Russell E. King, Dr. Henry L. W. Nuttle and Dr. Kristin A. Thoney, for serving on my Advisory Committee. Their time, patience and encouragement along the way is certainly appreciated.

I would like to especially thank Dr. Shu-Cherng Fang, who gave me a lifetime opportunity for studying at North Carolina State University. I would like to especially thank to Dr. Henry L. W. Nuttle, who always supported me, not only for this research, but also for all of my educational goals. I would like to especially thank to Dr. Russell E. King, who played the role as my advisor, training me for my first two years at NCSU. I would like to especially thank Dr. Kristin A. Thoney for her invaluable explanations of the Virtual Factory, and advice in computer programming.

I would like to thank all of the graduate students in the Hodgson/King research group for their support. I would also like to thank all my Thai friends in Raleigh for providing me support and a warming life.

I would like to thank my family. Special thanks go to my grandparents, Kit and Kin, my parents, Kovit and Suvimon Laoboonlur, and my sisters and brother, Bell, Best, Boom and Bye, who always provide unconditional love and infinite willingness to give me their best. Finally, special thanks to my fiancé, Thuanghatai Satranuvat, who has provided me with unlimited support and understanding.

Contents

List of Tables

Chapter 1

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Appendix A

List of Figures

Chapter 1

Chapter 5

Chapter 6

Chapter 1
Introduction

Dyeing and finishing are critical textile production processes. Dyeing is the process of adding colorants to fibers throughout the fabric, and finishing is the process of altering a fabric by changing its geometric and/or chemical nature. Dyeing and finishing are batch and multi-stage processes. Typically, each stage has parallel machines that may or may not be identical. A machine sequence (specified by an order) depends on the requirements for the finished fabric. There may be a number of alternative machine sequences for a given order. Setups of the machines are dependent on the production sequence of the orders and the capability of the machines. Dyeing and finishing processes are essentially flexible job shops with sequence dependent setups.

This research is concerned with developing a scheduling methodology for a knitted fabric dyeing and finishing plant. The plant production environment is as noted above. General scheduling objectives as well as business issues are addressed. Customers are classified into two groups: high priority customers who place large orders, and low priority customers who place small orders. There are two types of due dates that may be assigned to an order. One is the due date as given by the customer. On other orders the customer doesn't specify the due date. The company assigns its own due date for that order based on the customer's priority and the order size. The due date given by the customer is treated more strictly than one assigned by the company. The production environment is another issue. In the dyeing process, fabrics from the same order must be processed in the same type of machine. Rework rates are high in many processes. This research will address the real-world scheduling issues associated with dyeing and finishing, and develop a practical scheduling methodology for that scenario. A scheduling methodology will be developed based on the dyeing and finishing process flows, machines, and business processes of a case plant.

1.1 Overview of Dyeing and Finishing Process of the Case Plant

A general picture of production is gained by classifying the dyeing and finishing processes into six primary processes; 1) Dyeing, 2) Drying, 3) Cutting, 4) Surface Finishing, 5) Tubular Fabric Setting, and 6) Flat Fabric Setting. Production starts when greige fabric is dispatched from the warehouse to the dyeing area. After the fabrics are dyed, they are sent to the drying process. There are two types of drying machines: spinning and drying machines. Both types of machines are used to dry most all fabrics, starting with spinning. However, for very thin fabrics, only spinning machines are used. If an order requires the finished fabric in flat form, the tubular fabric is sent to the cutting (opening) machine before doing any finishing. If an order requires surface finishing, it is sent to the surface finishing section, and then to the setting machines. If the order doesn’t need surface finishing, it is sent directly to the setting machines. Flat fabrics are sent to a flat fabric setting machine. Tubular fabrics are sent to a tubular fabric setting machine. Packing is done after the setting process. Finished fabrics are sent to the finished fabrics warehouse for transport back to the customer. The flow of these primary processes is shown in figure 1-1.

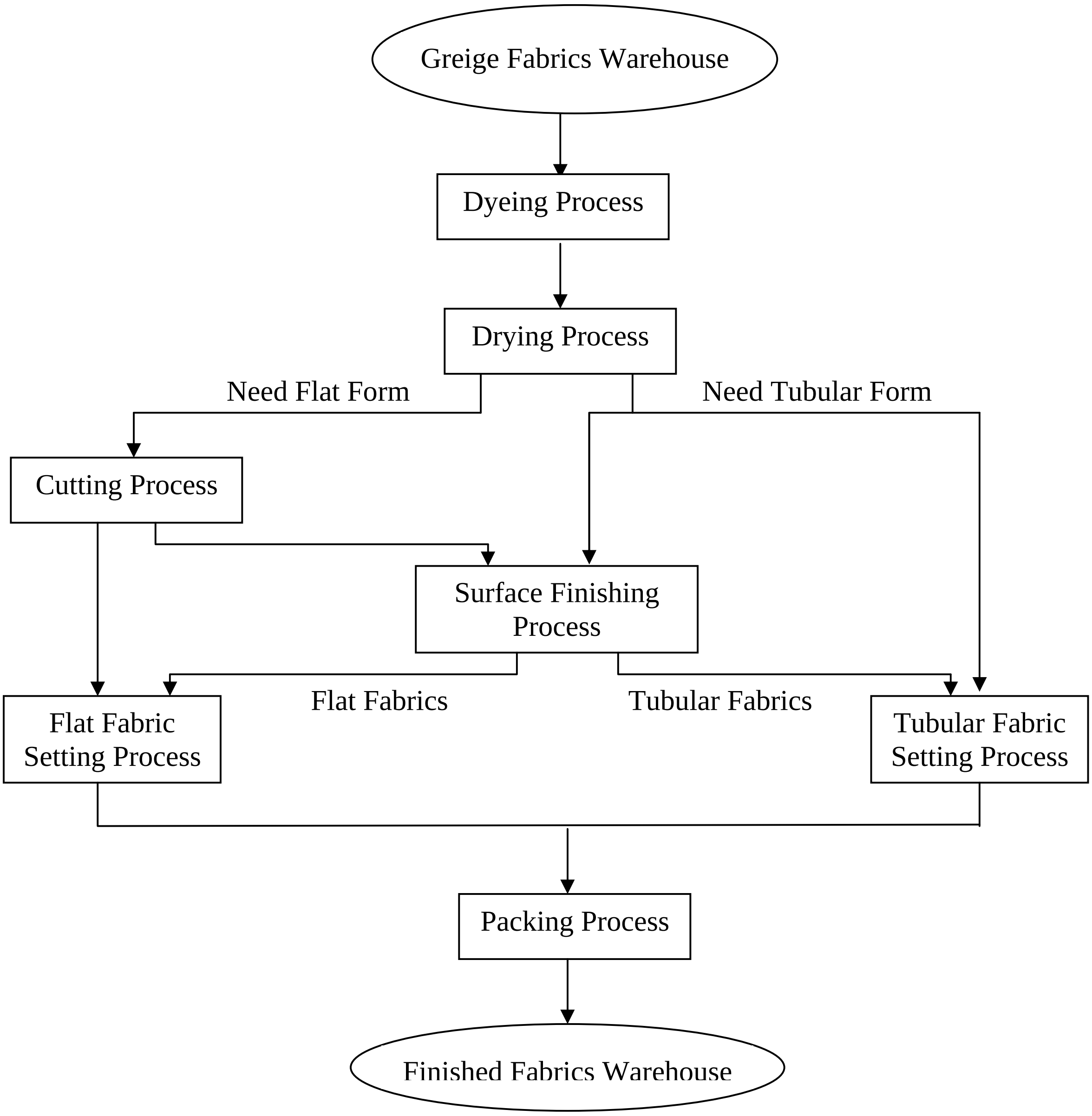


Figure 1-1: The Flow of Primary Processes in Dyeing and Finishing

1.1.1 Dyeing Process

Dyeing is the process of distributing colorant uniformly within the fabric, and also putting in chemicals to improve fabric characteristics. This process is performed by four different machine types: 1) O-Type High Temperature & High Pressure Dyeing Machine (O-HT); 2)

L-Type High Temperature & High Pressure Dyeing Machine (L-HT); 3) Rapid Winch Machine (RW); and 4) Winch Machine (W). Each type machine has various weight capacities. For O-HT and L-HT, both are high-temperature and high-pressure machines but have different operational details. The main difference between them is the operating temperature, since dyestuffs and chemicals need different temperatures in their reactions. HT, which has highest operating temperature, can be used for any process.

In selecting a machine, two things need to be determined: required operating temperature and batch size (total fabric weight in a batch). There is an additional concern in using HT machines. An order requiring two or more HT machines has to be processed in the same type of HT machine (either O-HT or L-HT), since processing an order in these two HT types gives different fabric characteristics, especially in fabric width.

1.1.2 Drying Processes

Drying is the process of drying the fabric before going to finishing. There are two sub-processes for this operation: spinning and drying.

The spinning process works just like a home washing machine. In spinning (hydro extractor) the major quantity of water is removed from the fabric by centrifugal force.

The drying process blows hot air through the fabric. There are two types of machines: a tube drying machine (vertical dryer), and a drying machine. There are two sizes of tube drying machine. In the tube dryer; fabric moves through the tubes of the machine in an open system. In the drying machine, fabric moves through the rollers in a closed system. Processing on the drying machine can be used to modify fabric structure by passing the fabric through a chemical bath before the fabric goes into the drying machine.

1.1.3 Cutting (Opening) Process

All greige fabrics delivered to the plant are in tubular form. If the customer needs the finished fabric in flat form, a cutting machine is used to convert the fabric from tubular to flat.

1.1.4 Finishing Processes

Finishing is the process of improving the fabric characteristics. Finishing is classified as chemical, and mechanical and/or thermal. A chemical or mixture of chemicals is applied to a fabric to impart desired characteristics such as water repellent flame retardant, bacteria and fungi resistant, and durable press. Mechanical and thermal finishing is a process in which the geometry of the fabric or the polymer arrangement within fibers has been altered. Fabric luster, smoothness, softness, hand, residual shrinkage and surface appearance are examples of properties that can be altered by mechanical/thermal finishing. There are three sub processes: wet finishing, surface finishing, and setting.

1.1.4.1 Wet Finishing (Softening) Process

Wet finishing is the process of passing the fabric through a bath containing finishing chemicals to improve fabric characteristics (Padder).

1.1.4.2 Surface Finishing Processes

Surface finishing is the process of improving the surface appearance and hand of the fabric. Surface finishing consists of different sequences of machines depending on the hair appearance requirement. Four types of process are used: raising, brushing, shearing and tumbling.

Raising is the process used to create a pile surface on a fabric. Fibers are deliberately pulled part way out of the yarn (by a raising machine) to give the fabric a hairy or fuzzy appearance and a soft surface texture. In this process, the order can be processed through multi-machine in the same area continuously instead of using only one machine. This increases the processing speed.

Brushing is also a process to raise fibers using a brushing machine, except that the density of the surface fibers is lower than the raising process.

Shearing is the process where the fabric is napped to obtain an even surface. Cutting the fiber ends to a uniform length. A shearing machine performs this process.

Tumbling is a napping process used to develop fabric texture (in a tumbling machine). Two types of surface finished fabric; anti-pilling fabric and woolen fabric, are processed in this machine.

1.1.4.3 Setting Processes

Setting is the process of drying fabrics to a specified width and placing the warp and filling yarns back at right angles to each other. In this manner, a fabric is adjusted to the desired width and straightened. Both tentering and heat setting processes are used for setting. Setting machines are separated into two types, a tubular fabric-setting machine and a flat fabric-setting machine (also called stenter), depending on the required finished fabric form. A tubular fabric setting machine can be classified in two types: a gas setting machine and a steam setting machine.

The complete process flow is shown in figure 1-2, and the type and number of machines in each process are shown in table 1-1.

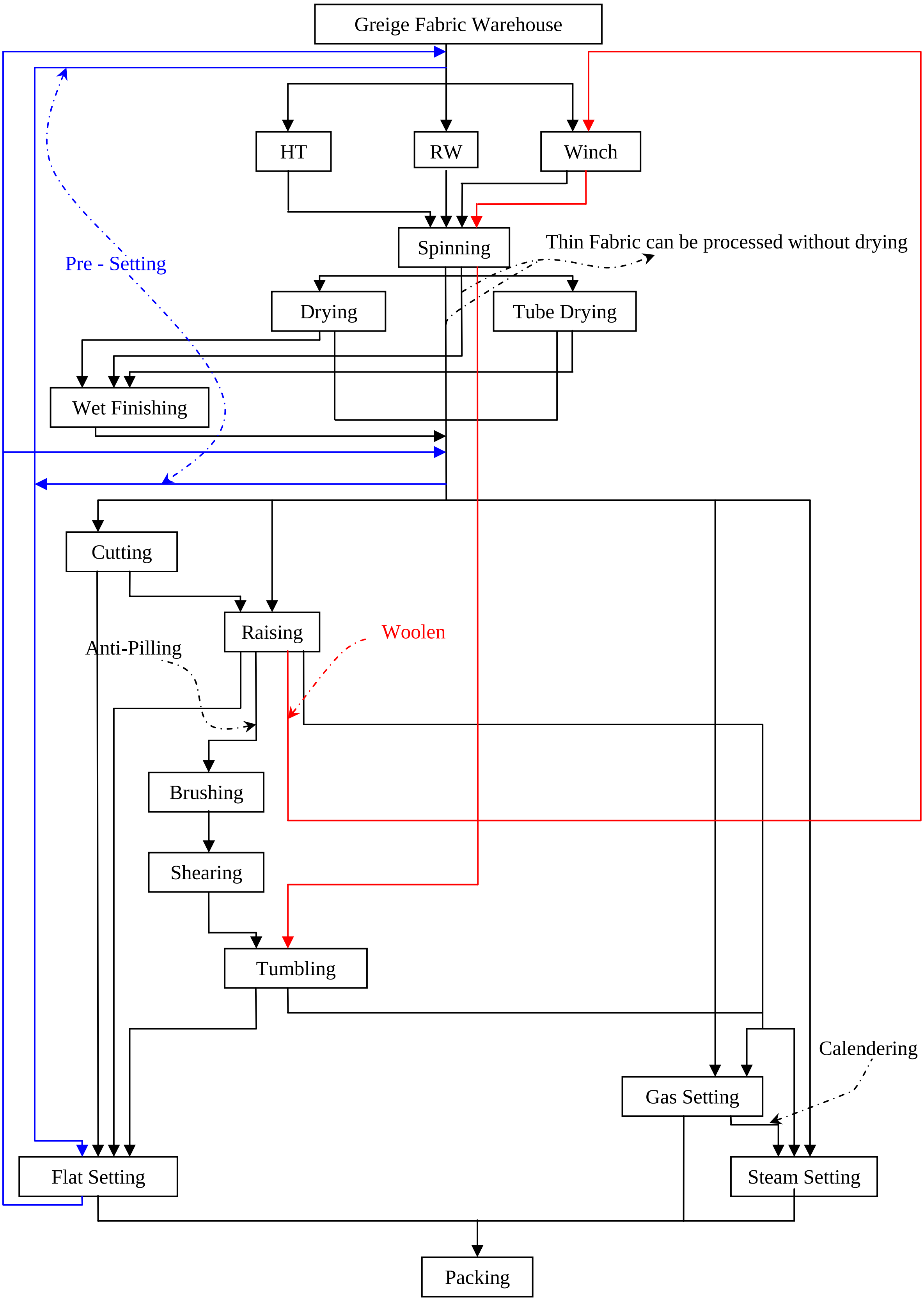


Figure 1-2: Complete Process Flow in Dyeing and Finishing

Table 1-1: Type and Number of Machines in Dyeing and Finishing Process

Process type	Machine Type	Capacity/Size	No. of Machines
Dyeing	High Temperature & Pressure	5 kg.	2
		50 kg.	1
		100 kg.	3
		200 kg.	4
		400 kg.	4
		600 kg.	3
		1000 kg.	3
		1200 kg.	1
	Rapid Winch	400 kg.	2
	Winch	40 kg.	1
		100 kg.	1
		200 kg.	1
Drying	Spinning		3
	Tube Drying	4 tubes	1
		6 tubes	2
	Drying		1
Cutting	Cutting		1
Wet Finishing	Wet Finishing		1
Surface Finishing	Raising	1.6 m.	5
		1.9 m.	5
		2.0 m.	4
		2.5 m.	1
	Shearing	2.0 m.	1
		2.1 m.	1
	Brushing		2
	Tumble		6
Setting	Gas Setting		3
	Steam Setting		2
	Flat Fabric Setting		2

1.2 Problem Description

In the dyeing and finishing production environment, the problem can be described as a (flexible) job shop scheduling problem with multiple products and sequence dependent setups. Each operation has parallel machines with both identical machines and non-identical machines.

1.2.1 Order Information

There are two classes of orders; high and low priority. Each order is assigned a sequence of operations, release time and due date. The processing time of an order on each machine is deterministic. The processing time is based on order size and dyeing and finishing requirements. In addition, the percentage of rework is quite high. There are two types of rework, major and minor. Major rework requires starting the process over (go back to an upstream process). Minor rework requires extension of the processing time in the current process.

1.2.2 Required Decisions

In the dyeing and drying processes, there may be a number of alternative machines that can be used for a given order. In the raising process, an operator needs to decide the number of machines used for an order. Another decision is the preempting availability of an order in finishing processes. Based on the complexity of machine conditions setting and quality required, some orders can be preempted. Rules for these decisions are complex and technical.

1.2.3 Scheduling Constraints

There are three scheduling constraints in the production of the case plant.

1. An order exceeding dyeing machine capacity must be separated equally to multiple dyeing machines. This requires that the processing conditions in all batches be the same.
2. O-HT and L-HT are both high pressure and high temperature dyeing machines, but they produce different fabric characteristics, especially with respect to fabric width. Therefore an order requiring two or more HT machines must be processed in the same type machine either O-HT or L-HT.
3. Major rework orders, which require starting over in an upstream process, are placed in the machine queue as a normal order.

1.2.4 Stochastic Unexpected Setups and Setup Times

Although setup matrices for each process are developed in this research for specifying setup time, unexpected setups (or surprise setups) may be occurred in actual operations because of unforeseen technological considerations.

1.2.5 Scheduling Objectives

Scheduling objectives are specified by the company business strategies. Short lead-time is one of the best marketing strategies. It aids in reducing greige fabric inventory and work-in-process. However, high priority customers are the most important (and majority) of the business. This requires that the company take care of them as a priority. Maximizing machine utilization is also desirable. Since most orders require the finished product to be flat fabric, supervisors tend to emphasize the flat fabric orders. This tends to make tubular fabric orders late, and causes idle time in the tubular fabric setting machines. The scheduling objectives will reflect these issues, which include both single and multiple objectives. Potential objectives to be studied include:

1. Minimize mean flow time;
2. Minimize maximum lateness;
3. Minimize maximum lateness while maximizing machine utilization;
4. Minimize maximum lateness of high priority customers subject to a maximum lateness of low priority customers;
5. Minimize maximum lateness of high priority customers subject to a maximum lateness of low priority customers, and maximize machine utilization

1.2.6 Dynamic Scheduling

The factory receives orders from customers every day. Some orders come later, but need to be finished earlier. The fabric for an order may be delivered to the plant in separate batches and separate days. To maintain consistent quality, reduce workload, and achieve the due date (if possible) these batches of fabric should be combined and processed together. In addition, rescheduling with new arrival orders may reduce the setup times of the current schedule. Processing major rework is another reason that production has to be rescheduled on a regular basis. Based on these production environments, this research will focus on dynamic scheduling (i.e., the scheduling approach can solve the industrial-sized problem with in the reasonable computational time.).

1.2.7 Scheduling Problem Variations

The scheduling problems can be modified by including the following interesting aspects.

- Various scheduling objectives in section 1.2.5
- Scheduling constraints in section 1.2.4
- Stochastic rework
- Stochastic unexpected setups and setup times
- Preemption availability

1.2.8 Assumptions

This study will focus only on developing a scheduling methodology that does not include the information complexity and decision making as noted above. Therefore, the following are information and decisions assumed to be known prior to scheduling.

1. Order released time and due date
2. Order priority
3. Order preempting availability
4. Order process sequence
5. Alternative machine types able to be used in each process for an order
6. Number of machines required in each process for an order
7. Order processing time for each process
8. Order stochastic rework rate

1.3 Research Objectives and Expected Scheduling System

The overall objective of this research is to first structure the actual industrial scheduling environment, and then to develop a scheduling system that can provide good schedules in a practical amount of computation time, for dyeing and finishing production. The specific objectives are to:

1. Develop the setup matrices for each process (dyeing and finishing).
2. Develop a scheduling methodology for the case plant.
3. Develop a simulation model that can be used to evaluate the scheduling methodology in various problem conditions.

1.4 Research Overview

Chapter 2 contains a review of literature related to the scheduling problem described. The main focuses are on scheduling with setup and job shop scheduling. In chapter 3, first setups in dyeing and finishing process are studied. Then setup matrices are developed. In chapter 4, the characteristics of the dyeing and finishing scheduling problem is analyzed and described. The scope of the scheduling problem solved in this research is also addressed in this chapter. The scheduling algorithm developed is discussed in chapter 5. Chapter 6 presents the experimental results and analysis for solving the scheduling problems. Chapter 7 is conclusions and a discussion of future research.

Chapter 2
Literature Review

2.1 Article Concentration

The problem studied in this research is a dyeing and finishing scheduling problem that involves multi-machines, multi-processes and sequence dependent setups. Thus the articles reviewed in this chapter concentrate on the multi-machine scheduling problem, including parallel machines, flow shop and job shop problems, with sequence dependent setups. Some interesting sequence independent setup problems and single machine problems are also reviewed.

Articles are classified in two main approaches; general scheduling and simulation based scheduling. In general scheduling, problems are solved by either optimization or heuristics. In the simulation approach, recursive simulation is used to develop schedules.

We begin with the general scheduling articles, which are grouped by shop environment and solution approach. Then simulation based scheduling is reviewed. Articles directly related to dyeing and finishing are reviewed separately and presented in the last section of this chapter.

2.2 Scheduling with Sequence Dependent Setups

Allahverdi, *et al.* [2] provide a comprehensive review of the literature on static scheduling problems involving setup considerations. They classify problems according to shop environments of single machine, parallel machine, flow shop and job shop; then into batch and non-batch jobs, and finally into sequence independent and dependent setup.

Panwalkar, *et al*. [36] conducted a survey to determine the extent of sequence dependent setups involved in industrial scheduling. From the survey, 75% of industrial managers said there was at least one operation requiring sequence dependent setup scheduling.

Krajewski, et al. [25] studied the impact of various factors in a production environment. They find that setup times and lot sizes have the greatest impact on reducing inventory and improving customer service.

Bruno and Downey [8] prove that the scheduling problem with deadline constraints and setup costs is NP-complete for even the case of a single machine with zero setup time and unit changeover costs.

2.2.1 Single Machine Problems

Using historical data for setup time and information of setup classification, White and Wilson [55] use multiple classification analysis to develop a prediction equation for estimating sequence dependent setup times. Then the relative values of the coefficients are used to develop a practical quantitative heuristic scheme for the manual sequencing of jobs based on the basic characteristics of setup operation of the machine. This procedure is applied in the single lathe machine problem to minimize setup time.

Taner [48] presents an n-job, N-family with n_i jobs in each family, single machine scheduling problem with sequence dependent family setups. A sequence of two neighborhood search strategies is developed to solve the problem for minimizing maximum lateness. In addition, the general scheduling rules, the variation in Earliest Due Date (*EDD*) rule and Shortest Setup Time (*STT*) rule, are studied in his research for the single machine dynamic scheduling problem with sequence dependent setup times. The study uses discrete time Markov decision processes to determine, which control rule is the best.

Monma and Potts [32] study the single machine problem and extend it to include batch setup times. Various optimality criteria, including maximum lateness, total weighted completion time, and number of the tardy jobs problems, are investigated. The study shows that once the order of jobs within each batch is known, a dynamic programming algorithm can be used to optimally merge the ordered batches into a single schedule.

2.2.2 Parallel Machine Problems

Monma and Potts [32] study the parallel machine model with and without preemption and show that the maximum completion time, maximum lateness, total weighted completion time, and number of tardy jobs problems are NP-hard, even for the case of two identical parallel machines and sequence independent setup times.

Optimization Approaches

Pearn, *et al*. [39] address the wafer probing scheduling problem (WPSP), which is a practical parallel machine scheduling problem. The WPSP case involves constraints on job clusters, job-cluster dependent processing time, due dates, machine capacity, and sequence dependent setup times. They formulate WPSP as an integer program and solve it using CPLEX. The objective in this case is to find a schedule that satisfies the due date restrictions without violating the machine capacity constraints, while minimizing the total machine workload.

Marsh and Montgomery [31] formulate a branch and bound algorithm to minimize the total setup times. In their study both identical and non-identical parallel machines problem are discussed. Jobs are assumed to be available at time zero have no due date restrictions.

Deane and White [13] use a branch and bound algorithm to solve parallel processors with sequence dependent setup. The objective is to minimize total setup costs subject to meeting workload-balancing restrictions. Their algorithm can be applied only to small or moderate size problems.

Kang *et al.* [23] present a sequence splitting technique for breaking down the scheduling problem into tractable sub-problems. A branch and bound algorithm is used to minimize the setup costs.

Ovacik and Uzsoy [34] study a special case of the parallel machine problem, where the setup times associated with each job is bounded from above by its respective processing time. They construct a worst case error bound for minimizing $L_{\max}$ and make span objectives.

Bitran and Gilbert [6] study a parallel machine problem with two magnitudes of sequence dependent setups. A network model is constructed to present the cost structure, which is used to define product families. A branch and bound algorithm is formulated for allocating the product families to minimize total setup costs. They also propose techniques to schedule the families in a given allocation and the products in a given family.

Heuristic Approaches

In the WPSP case, Pearn, *et al*. [39] propose an efficient procedure, called the Weighted-Saving algorithm, to solve the WPSP to near-optimally. Since only considering a combination with larger setup time saving may cause too much advancing in job starting time, the Weighted-Saving algorithm considers both the setup time savings and starting time slackness savings.

Franca, *et al*. [15] applied Tabu search to solve the multiprocessors problem of *n* jobs on *m* identical parallel processors with sequence dependent setup times. The objective is to minimize the execution time (make span). They propose a three phase heuristic to solve this problem. An initial phase constructs a starting solution, which is improved in the second phase by means of tabu search. The final phase attempts a further improvement on the busiest machine.

Tucci and Rinaldi [50] applied Tabu search to production scheduling in weaving (a textile process), which is in the class of the parallel non-identical machine scheduling with non-linear delay penalties and sequence dependent setup costs and times. The objective is to minimize make span, the number of wrap chain changes (setups) and total weighted tardiness.

2.2.3 Flow Shop Problems

Optimization Approaches

Corwin and Esogbue [11] present a dynamic programming approach for the two-machine flow shop problem, but only one machine, either first machine or second machine, is characterized by sequence dependent setup times. The objective is to minimize the make span subject to meeting certain due date constraints.

Srikar and Ghosh [44] formulate mixed integer linear programming model for non-identical *n*-job, *M*-stage sequence dependent flow shop with the objective of make span minimization.

Stafford and Tseng [45] report some corrections on the model proposed by Srikar and Ghosh [44]. They also present three new models to solve the same problem.

Uskup and Smith [51] study a two-stage production-sequencing problem with due date constraints and sequence dependent setup times that are proportional to the processing times. Branch and bound is used to minimize the total setup cost without violating the due date constraint. In this study, two stations can have different processing orders.

Bianco *et al.* [5] show that the flow shop scheduling problem with sequence dependent setup times is equivalent to the TSP with additional visiting time constraints. They present a mathematical formulation and two lower bounds, and two heuristic algorithms.

Heuristic Approaches

Das, *et al*. [12] develop a savings index heuristic for the permutation flow shop scheduling problem with sequence dependent setup times to find an approximately minimum make span.

Gupta and Darrow [17] show that a two-machine flow shop scheduling problem with the objective of make span minimization is NP-complete in the presence of sequence dependent setups. They also show that permutation schedules for this problem may lead to suboptimal solutions. In this study they present four approximate algorithms. However, two of them become ineffective when the setup times are relatively equal to the processing times.

Szwark and Gupta [47] present a polynomial bounded approximate method for formulating permutation schedules to minimize the make span. They show that the optimal solution is found in the two-machine case.

Mercado and Bard [41] apply an enhanced heuristic to transform flow shop problem into a TSP problem with a cost function that penalizes for large setup times and bad fitness of schedule. The objective is to minimize the makespan.

Rajendra and Ziegler [40] present a heuristic procedure and supplementary improvement scheme to minimize the weighted flow times of a static flow shop with sequence dependent setups.

Pathasarty and Rajendra [37] use a simulated annealing procedure with a neighborhood structure based on a random insertion perturbation scheme to minimize the mean weighted tardiness of a flow shop with sequence dependent setup times.

Pathasarty and Rajendra [38] extend their problem in [37]. The additional objectives of minimizing the maximum weighted tardiness and minimizing the total tardiness are solved in this study.

2.2.4 Job Shop Problems

Kim and Bobrowski [24] investigate the dynamic job shop scheduling problem that is complicated by sequence-dependent-setup up times. Several scheduling rules are tested to determine the effect of setup time and/or due date information. A simulation model, using SLAM II, of a nine machines job shop is used for experimentation. The study shows that setup time must be given explicit consideration in the scheduling decision when it is sequence-dependent. Also, the due date information should be included in the sequencing decision in order to improve due date performance.

Optimization Approach

Gupta [16] apply branch and bound to solve static scheduling problems involving *n*-jobs and *m*-machines with sequence dependent setup times. The objective is to minimize setup cost.

Hwang and Sun [21] address a job shop scheduling problem with setup dependence where a set of *n* jobs needs to be scheduled on two machines for the side frame press shop in a truck manufacturing company. A characteristic of this real life shop scheduling problem is two-step-prior job dependent setup times. A dynamic programming approach is used to solve the problem with the objective of minimizing makespan.

Choi and Korkmaz [10] formulate a mixed integer model to minimize the makespan in a flexible manufacturing cell with multiple jobs, multiple machine stations, and sequence dependent setup times. Their model gives better solutions than Zhou and Egbelu [56] and is solvable in a polynomial time.

Heuristic Approach

In the same problem of Hwang and Sun [21] mentioned above, develop a genetic algorithm (GA) to get near-optimal solutions within a reasonable time.

Tucci and Rinaldi [50] compare GA, Tabu search (TS) and simulated annealing (SA) in a particular problem of flexible job shop. They found that Tabu search was systematically superior in terms of solution quality. In addition, though a straight comparison was not possible, computational times with TS were at least one tenth of those achieved using both SA and GA.

Flynn [14] shows that applying sequence dependent setup scheduling procedures and group technology principles can increase the output capacity of a cellular manufacturing shop.

Jensen, *et al.* [22] present several family-based dispatching rules for a job shop that contains jobs classified into family having family setup times. Simulation is used and the results show the performance improvement in the studied scheduling objectives, especially when the ratio of family setup times to processing times is greater.

Mahmoodi, *et al.* [29] also used simulation to study family-based dispatching rules, and refer to this problem as group scheduling. The empirical results show significant performance improvement in scheduling.

Zhou and Egbelu [56] develop a heuristic algorithm to minimize the makespan in a flexible manufacturing cell with multiple jobs, multiple machine stations, and sequence dependent setup times. Their algorithm generates an initial solution and then is modified by an expert.

Sun and Noble [46] decompose a job shop problem with release times, due dates, and sequence dependent setups constraints into a series of single machine problems and schedule the machine one at a time. The criticality of each machine is determined based on its marginal contribution to the overall objective function. The single machine problems are solved using Lagrangian relaxation and shifting bottleneck approaches.

Independence Setup

Some interesting job shop with sequence independent setup literature is reviewed here.

Warner and Winkler [54] present a combination constructive algorithm for creating an initial feasible solution, and then apply a local improvement algorithm for solving the job shop makespan problem. They also take advantage from the structure of the makespan objective, which has a small neighborhood space, in only searching neighbors of the critical path.

Adams, *et al.* [1] present a shifting bottleneck procedure for solving the minimum makespan scheduling problem without setups.

Balas, *et al.* [4] use the shifting bottleneck procedure for solving the $L_{\max}$ job shop problem.

2.3 Simulation Based Scheduling Problems

A recursive simulation technique is one of the interesting approaches for solving scheduling problems. The virtual factory approach keeps rescheduling the sequence with such estimated information from simulation, until the certain criteria are met. Lawrence and Morton [27] are the first who proposed this approach.

Vepsalainen and Morton [52] develop the iterative myopic dispatching method for solving the weighted tardiness job shop problem. A deterministic simulation technique is applied for estimating the lead time using in the next iteration. The iterative process is terminated when criteria are met.

Ovacik and Uzsoy [28] apply a modified dispatching rule, which determines both current jobs in the queue and soon arrival. In this case, they develop a deterministic simulation technique to predict the arrival of a job at a machine. Various problems are studied with this approach including job shop with sequence dependent setups.

Hodgson, *et al*. [18, 19] present a simulation based approach for the multi-machine job shop scheduling problem for minimizing the maximum lateness. This approach is applied to large job shop environments and is able to provide optimal, or near optimal schedules under certain conditions.

Taner [48] incorporates job shop scheduling problem with sequence dependent setup time considerations into the Virtual Factory System. This simulation system gives estimates of time and due date information for each job on each machine. The resulting sub-problem is a single machine problem. The scheduling objective is to minimize the maximum lateness of all jobs.

Weintruab *et al.* [53] develop a Tabu search algorithm used with the virtual factory to evaluate alternative sequencing, routings, and operations. They show that alternative operation provides the most useful, and the least in alternative sequences.

2.4 Dyeing and Finishing Scheduling Problems

Values of color characteristics in a measurement system, called CIELAB, are applied by Morales, *et al*. [33] to develop a color difference matrix. Then they consider a single machine problem in dyeing as an asymmetric traveling salesman problem. A Branch and bound is used to solve this problem with the objective of finding a sequence of colors to produce the minimum total difference between the desired color and the obtained color.

Using the same color difference matrix and the same scheduling problem as Morales, *et al*. [15], Maldonado, *et al*. [30] applied a search heuristic to solve the problem. However, the efficiency of the method depends on the goodness of the initial solution. They use a sequence proposed by experts for this purpose.

Livingston and Sommerfeld [28] use GPSS to model and perform a discrete-event simulation of a textile dyeing and finishing mill. The model is used to determine the effects of market demands, maintenance practices, quality control policies, and total production on equipment and manpower utilization, work-in-process inventory, and total processing time.

2.5 Survey Comment

Single machine problems have received the most attention in sequence dependent setups due to their relative simplicity. Only a few researchers have studied multi-machine and/or multi-stage scheduling problems with sequence dependent setups. A couple of papers consider stochastic problems with setup times. Virtually all approaches involve the use of heuristic procedures due to the complexity of the problem. Optimization can be used in only small problems. In addition, the majority of papers considers the makespan, and do not consider due date related objectives. Only a few papers present scheduling problems related to dyeing and finishing scheduling directly.

Chapter 3
Setups in Dyeing and Finishing Process

With multiple products and multiple sequence dependent setups in the dyeing and finishing process, studying the relationship between fabric types, order requirements (e.g. color, finishing characteristics), order size, and setup time are critical for developing a scheduling method. This relationship provides the information that determines when a setup is necessary. Thus, one of the objectives of this research is to develop setup matrices for scheduling dyeing and finishing processes. The setup matrices are developed based on information and experience of the case plant.

In general, all machines in the dyeing and finishing processes have sequence dependent setups. However, some machines require a very short time in their setup. These machines will be considered to have setup independence.

The machines requiring the long setup times will be studied for developing the setup time matrix. Some machines have more than one setup, so the setups have to be ranked the priority based on the setup times required. However, some setups require a very short time, and are also considered as sequence independent. The setup type, sequence dependence and priority in each machine are summarized in table 3-1. The fabric type abbreviations used are described in appendix B.

Table 3-1: Summary of Machine Setup in Dyeing and Finishing Process

Process Type	**Machine Type**	**Setup Type**	**Dependence & *Independence***	**Priority**
Dyeing	High Temperature & High Pressure	Cleaning	Dependence	
	Rapid Winch	Cleaning	Dependence	
	Winch		*Independence*	
Drying	Spinning	Cleaning	*Independence*	
	Tube Drying	Major Cleaning	Dependence	1
		Minor Cleaning	Dependence	2
	Drying	Major Cleaning	Dependence	1
		Minor Cleaning	Dependence	2
Cutting	Cutting	Cleaning	*Independence*	
Wet Finishing	Wet Finishing		*Independence*	
Surface Finishing	Raising	Major Cleaning	Dependence	1
		Minor Cleaning	Dependence	2
		Equipment Setting	*Independence*	
	Shearing	Major Cleaning	Dependence	1
		Minor Cleaning	Dependence	2
		Equipment Setting	*Independence*	
	Brushing		*Independence*	
	Tumble	Major Cleaning	Dependence	1
		Minor Cleaning	Dependence	2
Setting	Gas Settingsd	Major Cleaning	Dependence	1
		Minor Cleaning	Dependence	2
		Width Tool Changing	Dependence	3
		Cooling	Dependence	4
		Heating	Dependence	5
		Width Adjusting	Dependence	6
	Flat Fabric Setting	Major Cleaning	Dependence	1
		Minor Cleaning	Dependence	2
		Cooling	Dependence	3
		Heating	Dependence	4
		Width Adjusting	*Independence*	
	Steam Setting		*Independence*	

3.1 Color Groups, Color Shade Difference and Color Migration Effect

Before going into the detail of developing the setup matrices, the concepts of color groups, color shade difference and color migration effect need to be understood.

3.1.1 Color Groups

Based on the general concept of the case plant, the colors are grouped into three main groups; light, medium, and dark. The ranges of total percentage of color in a formula used as the standard in classifying the groups for each type of fabric are shown in table 3-2. The laboratory technicians of the case plant specify this standard. However, in some processes, white is determined as separate from the light color, because of different setup requirements.

Table 3-2: Color Groups

Fabric Type			
Poly, Cotton, TC, and CVC		***Nylon***	
Total Color Percentage	**Color Group**	**Total Color Percentage**	**Color Group**
< 0.6%	Light	< 0.3%	Light
0.6% - 1.5%	Medium	0.3% - 0.8%	Medium
> 1.5%	Dark	> 0.8%	Dark
Contain only OBA colors or bleaching chemicals	White	Contain only OBA colors or bleaching chemicals	White

3.1.2 Color Shade Difference

To explain this topic, we have to distinguish between fabric colors and colors in a formula. A color formula for a fabric may include one or more colors combined. From here on, the term “*color*’ means the fabric color; and the term “*color element*” means color used in writing a formula. The shade difference of two colors is determined by comparing the color elements in two formulas. There are two cases that two colors have the same shade.

- Two color formulas contain the same color elements, and the percentage used for each color element in each formula has a difference less than 10%.
- Two color formulas contain the same color elements. All color elements in a formula are greater (less) than another one, and the amount of increase (decrease) in each color element has a difference less than 10%.

Otherwise, two colors have shade difference.

3.1.3 Color Migration Effect

When processing some dark colors at high temperature in the drying and finishing processes, dyestuff can be evaporated from the fabric and contaminate the machine. This is called thermo-migration effect and is one of the major factors requiring the machine setup. Normally, dyestuffs are classified into two classes, regular and special. Only regular dark color has this effect.

3.2 Setups in Dyeing and Finishing Machines

All setup matrices discussed in this section are shown at the end of the chapter.

3.2.1 Setup in Dyeing Machine

After dyeing a fabric, most of the dyestuff used becomes an integral part of the fabric, but a fraction remains in the bath and in the dyeing equipment. The unused fraction represents a contaminating element, which may affect subsequent dyeing processes. The type of fabric and the color darkness of the subsequent order are the factors that determine the setup requirement. The color used for a type of fabric may not contaminate some types of fabric.

Thus, if the subsequent fabric type can be contaminated by the current color used, and has a lighter color, then a machine cleaning before processing is required.

Two setup matrices for the dyeing machine are presented. The first matrix (table 3-4) determines the setup between processing two fabrics having different yarn types. The second matrix (table 3-5) determines the setup between processing two fabrics having same yarn types by considering color difference.

3.2.2 Setup in Tube Drying Machine and Drying Machine

The tube drying machine and drying machine is only used to dry the fabric with low temperature. There is just a little color contamination on the machine. Therefore the machine can process two consecutive orders having quite different color without setup. Color groups, thermo-migration effect, and color shade difference are factors used to determine a setup. A setup has to be determined only when processing an order next to the dark color group and can be classified into two cases based on with or without the thermo-migration effect. For dark color without thermo-migration effect, a minor cleaning is required when processing the light color group after the dark color group. For the dark color group with the thermo-migration effect, processing light color, medium color, and different shade dark color next to it, a major cleaning is required. The setup matrix for these machines is shown in table 3-6.

3.2.3 Setup in Raising Machine

There are two types of setup in these machines: equipment changing, and machine cleaning. The equipment has to be reset based on the surface finish requirement. Since each order has a different surface requirement, the equipment has to be set for every order. Thus, this setup is considered to be sequence independent. For machine cleaning, there are two approaches, major and minor, depending on how dirty the machine is.

There are two general cases requiring machine cleaning (table 3-7). The first case is when the short pile fabric (thin surface fabric) order has to be processed after a long pile fabric (thick surface fabric) order; the machine needs to be cleaned before processing the short pile order. The second case is that machine cleaning has to be done before a light or medium color order processed next to a dark color order. To apply these two rules, the fabric types, color shade difference, and batch sizes of two consecutive orders have to be included in the decision. Fabrics types are categorized into 3 groups:

1. Poly and TK with thick surface finishing
2. Cotton, CVC, TC with thick surface finishing
3. All fabric types with thin surface finishing, and other fabric types, not poly, TK, cotton, CVC, and TC, with thick surface finishing

However, if the size of the subsequent order is less than 20% of the current order, the effect of current order still applies to the next subsequent order.

These two setups may be needed at the same time. They can be done in parallel, and the cleaning time, which is the longer, represents the setup time.

3.2.4 Setup in Shearing Machine

The shearing machine has the same setup types as the raising machine (e.g., equipment changing and machine cleaning), but machine cleaning has only one type, major cleaning. Equipment changing is again considered as a sequence independent setup.

The cleaning is required when the subsequent order has different color group, and/or different shade color. In addition, for only this machine, white is determined to be separate from the light color group for setup consideration as shown in table 3-8. However, if the subsequent order is less than 20% of the current order, the effect of current order still applies to the next subsequent order.

3.2.5 Setup in Tumble Machine

Since there are only two surface finishing types; anti-pilling finishing and woolen finishing; processed in the tumble machine, switching between them is the main factor requiring setup. A major cleaning is required when the machine is switched from woolen finishing to anti-pilling finishing. For switching from anti-pilling finishing to woolen finishing, a major cleaning is required only when the color is changed from dark to light. A minor cleaning is required when, considering two orders in the same finishing types, two consecutive orders have much different color and/or color shade. The cleaning for this machine is shown in table 3-9.

3.2.6 Setup in Gas Setting Machine

There are five setups in this machine; machine cleaning, width equipment changing, cooling, heating, and width equipment adjusting; and their priorities are ranked respectively. The machine cleaning requirement (table 3-10) is determined based on the same concept as the tube drying machine and drying machine. If the subsequent order is less than 20% of the current order, the effect of current order still applies to the next subsequent order.

In width setting equipment, a size of the width setting equipment can be used for a specific range of width as shown in table 3-3. If the new order requires a width out of the range of the current equipment, it needs to be changed manually to the proper size. If the new order requires a different size from the current one, but still is in the width range of the equipment, it just needs to be adjusted.

Table 3-3: Range of Width Equipment in Gas Setting Machine

Width Range (inches)	No. of equipment
15 - 20	1
22 - 34	1
28 - 38	1
35 - 50	1
40 - 60	2

When processing a new order, the setting temperature has to be cooled down or heated up to the new proper level. Generally, fabrics with the same type, color type (e.g. regular or special), and color group are processed in the same temperature range. However, some colors, which have significant thermo-migration effect, are processed at different temperature ranges from their color group. The cooling time required is longer time than heating up.

More than one setup may be required at the same time. However, some setups can be done in parallel, such as temperature setting and cleaning, or temperature setting and width equipment adjusting. Therefore, the maximum setup time of parallel setups is used to determine the setup time.

3.2.7 Setup in Flat Fabric Setting Machine (Stenter)

There are four setups in this machine; machine cleaning, cooling, heating, and width equipment adjusting. Since width equipment can be done quickly, it is considered to be a sequence independent setup. Machine cleaning is ranked as the highest priority and cooling and heating respectively. The temperature-setting requirement is determined based on the same concept used with gas setting machine. The difference is the cooling time required is *much* longer than the heating time.

The machine-cleaning requirement (table 3-11) is more complex than the gas setting machine. Color used with a fabric type contaminates other fabric types at the different levels. Thus fabric types are categorized in 4 groups.

1. Cotton, rayon, and nylon
2. Poly and TK
3. CVC, and TC
4. Others

They are used to determine the cleaning requirement for processing the same or different fabric types consecutively. Not only the fabric type, but also the color group, color shade, and thermo-migration effect are determined mutually. Moreover, dark colors with thermo-migration and red color shade may need to be processed subsequently with another dark color with thermo-migration effect and red color shade depending on the fabric type. Major cleaning is used by dark color with thermo-migration effect and minor cleaning by dark color without thermo-migration effect. However, if the subsequent order is less than 20% of the current order, the effect of the current order still applies to the next subsequent order.

3.3 Comments on the Dyeing and Finishing Setup Matrices

The dyeing and finishing matrices developed are used for scheduling purposes of this research only. They are not the universal knowledge, and were developed based on the experience and knowledge of the case plant. The expectation is to apply these matrices with the developed scheduling methodology for this specific plant. Furthermore, since there is huge variation in product and process, the matrices are not 100% applicable or correct. In the real situation, some case may not follow the rules. Therefore, stochastic unexpected setups should be studied.

Table 3-4: Dyeing Machine Setup Matrix1 for Two Fabrics Having Different Yarn Types

		Cotton			**Poly**			**TC, CVC**			***Fabric type***
		Light	**Medium**	**Dark**	**Light**	**Medium**	**Dark**	**Light**	**Medium**	**Dark**	***Color***
Cotton	**Light**	▨	▨	▨	-	-	-	-	-	-	
	Medium	▨	▨	▨	-	-	-	-	-	-	
	Dark	▨	▨	▨	🟨	-	-	🟨	-	-	
Poly	**Light**	-	-	-	▨	▨	▨	▨	▨	▨	
	Medium	-	-	-	▨	▨	▨	▨	▨	▨	
	Dark	🟨	🟨	🟨	▨	▨	▨	▨	▨	▨	
TC, CVC	**Light**	▨	▨	▨	-	-	-	▨	▨	▨	
	Medium	▨	▨	▨	-	-	-	▨	▨	▨	
	Dark	▨	▨	▨	🟨	-	-	▨	▨	▨	
Fabric type	***Color***										

▨ Determine the setup requirement follow the dyeing setup matrix 2

🟨 Require machine cleaning

Table 3-5: Dyeing Machine Setup Matrix 2 for Two Fabrics Having Same Yarn Type

		Off White	**Bleach**	**OBA**	***Color***
		Any			***% Color***
Off White		-	-	-	
Bleach		🟨	-	-	
OBA		🟨	-	-	
Cream	**< 0.6 %**	-	-	-	
Yellow		🟨	🟨	🟨	
Pink		🟨	🟨	🟨	
Blue		🟨	🟨	🟨	
Beige		🟨	🟨	🟨	
Brown		🟨	🟨	🟨	
Purple		🟨	🟨	🟨	
Green		🟨	🟨	🟨	
Gray		🟨	🟨	🟨	
Blue	**0.6 % - 1.5 %**	🟨	🟨	🟨	
Orange		🟨	🟨	🟨	
Pink		🟨	🟨	🟨	
Yellow		🟨	🟨	🟨	
Brown#2		🟨	🟨	🟨	
Green#2		🟨	🟨	🟨	
Green#3		🟨	🟨	🟨	
Navy	**> 0.8 %**	🟨	🟨	🟨	
Black		🟨	🟨	🟨	
Red #2		🟨	🟨	🟨	
Green#4		🟨	🟨	🟨	
Brown		🟨	🟨	🟨	
Purple#2		🟨	🟨	🟨	
Gray		🟨	🟨	🟨	
Blue		🟨	🟨	🟨	
Purple		🟨	🟨	🟨	
Color	***% Color***				

🟨 Cleaning

Table 3-5: Dyeing Machine Setup Matrix 2 for Two Fabrics Having Same Yarn Type (continued)

		Cream	**Yellow**	**Pink**	**Blue**	**Beige**	**Brown**	**Purple**	**Green**	**Gray**	***Color***
		< 0.6 %									***% Color***
Off White		-	-	-	-	-	-	-	-	-	
Bleach		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
OBA		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Cream	**< 0.6 %**	-	-	-	-	-	-	-	-	-	
Yellow		-	-	-	-	-	-	-	-	-	
Pink		-	-	-	-	-	-	-	-	-	
Blue		🟨	🟨	-	-	-	-	-	-	-	
Beige		🟨	🟨	-	-	-	-	-	-	-	
Brown		-	-	-	-	-	-	-	-	-	
Purple		-	-	-	-	-	-	-	-	-	
Green		-	-	-	-	-	-	-	-	-	
Gray		-	-	-	-	-	-	-	-	-	
Blue	**0.6 % - 1.5 %**	🟨	🟨	🟨	-	-	-	-	-	-	
Orange		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Pink		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Yellow		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Brown#2		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Green#2		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Green#3		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Navy	**> 0.8 %**	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Black		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Red #2		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Green#4		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Brown		🟨	🟨	🟨	🟨	🟨	-	🟨	🟨	🟨	
Purple#2		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Gray		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Blue		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Purple		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Color	***% Color***										

🟨 Cleaning

Table 3-5: Dyeing Machine Setup Matrix 2 for Two Fabrics Having Same Yarn Type (continued)

		Blue	Orange	Pink	Yellow	Brown#2	Green#2	Green#3	*Color*
		0.6 % - 1.5 %							*% Color*
Off White		-	-	-	-	-	-	-	
Bleach		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
OBA		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Cream	< 0.6 %	-	-	-	-	-	-	-	
Yellow		-	-	-	-	-	-	-	
Pink		-	-	-	-	-	-	-	
Blue		-	-	-	-	-	-	-	
Beige		-	-	-	-	-	-	-	
Brown		-	-	-	-	-	-	-	
Purple		-	-	-	-	-	-	-	
Green		-	-	-	-	-	-	-	
Gray		-	-	-	-	-	-	-	
Blue	0.6 % - 1.5 %	-	🟨	🟨	🟨	-	-	🟨	
Orange		🟨	-	-	-	-	-	🟨	
Pink		🟨	-	-	🟨	-	-	🟨	
Yellow		🟨	-	🟨	-	-	-		
Brown#2		🟨	🟨	-	🟨	-	-	🟨	
Green#2		🟨	🟨	🟨	🟨	🟨	-	🟨	
Green#3		🟨	🟨	🟨	🟨	🟨	🟨		
Navy	> 0.8 %	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Black		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Red #2		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Green#4		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Brown		🟨	-	🟨	🟨	-	-	🟨	
Purple#2		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Gray		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Blue		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Purple		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Color	*% Color*								

🟨 Cleaning

Table 3-5: Dyeing Machine Setup Matrix 2 for Two Fabrics Having Same Yarn Type (continued)

		Navy	Black	Red #2	Green#4	Brown	Purple#2	Gray	Blue	Purple	*Color*
		> 0.8 %									*% Color*
Off White		-	-	-	-	-	-	-	-	-	
Bleach		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
OBA		🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Cream	**< 0.6 %**	-	-	-	-	-	-	-	-	-	
Yellow		-	-	-	-	-	-				
Pink		-	-	-	-	-	-				
Blue		-	-	-	-	-	-	-	-	-	
Beige		-	-	-	-	-	-	-	-	-	
Brown		-	-	-	-	-	-	-	-	-	
Purple		-	-	-	-	-	-	-	-	-	
Green		-	-	-	-	-	-	-	-	-	
Gray		-	-	-	-	-	-	-	-	-	
Blue	**0.6 % - 1.5 %**	-	-	-	-	-	-	-	-	-	
Orange		-	-	-	-	-	-	-	-	-	
Pink		-	-	-	-	-	-	-	-	-	
Yellow		-	-	-	-	-	-	-	-	-	
Brown#2		-	-	-	-	-	-	-	-	-	
Green#2		-	-	-	-	-	-	-	-	-	
Green#3		-	-	-	-	-	-	-	-	-	
Navy	**> 0.8 %**	-	-	🟨	-	🟨	-	-	🟨	🟨	
Black		-	-	🟨	🟨	🟨	🟨	🟨	🟨	🟨	
Red #2		-	-	-	🟨	-	-	-	🟨	-	
Green#4		-	-	-	-	-	-	-	-	-	
Brown		-	-	-	-	-	-	-	-	🟨	
Purple#2		-	-	-	-	-	-	-	-	-	
Gray		-	-	🟨	-	-	-	-	-	-	
Blue		-	-	-	-	-	-	-	-	-	
Purple		-	-	-	-	-	-	-	-	-	
Color	*% Color*										

🟨 Cleaning

Table 3-6: Drying Machine and Tube Drying Machine Setup Matrix

		Light	**Medium**	**Dark**		***Color***
		Any	**Any**	**Same**	**Different**	***Shade***
Light	**No**	-	-	-	-	
Medium	**No**	-	-	-	-	
Dark	**No**	▨	-	-	-	
	Yes	🟨	🟨	-	🟨	
Color	***Migration***					

🟨 Major Cleaning

▨ Minor Cleaning

Table 3-7: Raising Machine Setup Matrix

				Poly, TK						*Fabric Type*
				Thick						*Pile Thickness*
				Light		Medium		Dark		*Color*
				Same	Different	Same	Different	Same	Different	*Shade*
Poly, TK	**Thick**	**Light**	**Any**	-	-	-	▨	▨	▨	
		Medium	**Any**	▨	▨	-	-	▨	▨	
		Dark	**Any**	▨	▨	-	▨	-	-	
Cotton, CVC, TC	**Thick**	**Light**	**1-50**	-	-	▨	▨	🟨	🟨	
			51 and up	-	-	🟨	🟨	🟨	🟨	
		Medium	**1-50**	▨	▨	-	-	▨	▨	
			51 and up	🟨	🟨	-	-	🟨	🟨	
		Dark	**1-50**	🟨	🟨	▨	▨	-	▨	
			51 and up	🟨	🟨	🟨	🟨	-	🟨	
Others	**Any**	**Light**	**Any**	-	-	-	-	-	-	
		Medium	**Any**	▨	-	-	-	-	-	
		Dark	**Any**	🟨	🟨	-	▨	-	-	
Fabric Type	*Pile Thickness*	*Color*	*No. of Roll*							

▨	Minor Cleaning
🟨	Major Cleaning

Table 3-7: Raising Machine Setup Matrix (continued)

				Cotton, CVC, TC						*Fabric Type*
				Thick						*Pile Thickness*
				Light		Medium		Dark		*Color*
				Same	Different	Same	Different	Same	Different	*Shade*
Poly, TK	**Thick**	**Light**	**Any**	-	-	▨	▨	▨	▨	
		Medium	**Any**	▨	▨	-	-	▨	▨	
		Dark	**Any**	▨	▨	▨	▨	-	▨	
Cotton, CVC, TC	**Thick**	**Light**	**1-50**	-	-	▨	▨	🟨	🟨	
			51 and up	-	-	🟨	🟨	🟨	🟨	
		Medium	**1-50**	▨	▨	-	-	▨	▨	
			51 and up	🟨	🟨	-	-	🟨	🟨	
		Dark	**1-50**	🟨	🟨	▨	▨	-	▨	
			51 and up	🟨	🟨	🟨	🟨	-	🟨	
Others	**Any**	**Light**	**Any**	-	-	-	-	-	-	
		Medium	**Any**	▨	-	-	-	-	-	
		Dark	**Any**	🟨	🟨	-	▨	-	-	
Fabric Type	*Pile Thickness*	*Color*	*No. of Roll*							

▨	Minor Cleaning
🟨	Major Cleaning

Table 3-7: Raising Machine Setup Matrix (continued)

				Others						*Fabric Type*
				Any						*Pile Thickness*
				Light		Medium		Dark		*Color*
				Same	Different	Same	Different	Same	Different	*Shade*
Poly, TK	**Thick**	**Light**	**Any**	-	-	▨	▨	▨	▨	
		Medium	**Any**	▨	▨	-	-	▨	▨	
		Dark	**Any**	▨	▨	▨	▨	-	▨	
Cotton, CVC, TC	**Thick**	**Light**	**1-50**	-	-	▨	▨	🟨	🟨	
			51 and up	-	-	🟨	🟨	🟨	🟨	
		Medium	**1-50**	▨	▨	-	-	▨	▨	
			51 and up	🟨	🟨	-	-	🟨	🟨	
		Dark	**1-50**	🟨	🟨	▨	▨	-	▨	
			51 and up	🟨	🟨	🟨	🟨	-	🟨	
Others	**Any**	**Light**	**Any**	-	-	-	-	-	-	
		Medium	**Any**	▨	-	-	-	-	-	
		Dark	**Any**	🟨	🟨	-	▨	-	-	
Fabric Type	*Pile Thickness*	*Color*	*No. of Roll*							

▨	Minor Cleaning
🟨	Major Cleaning

Table 3-8: Shearing Machine Setup Matrix

	White	**Light**		**Medium**		**Dark**		***Color***
	Any	**Same**	**Different**	**Same**	**Different**	**Same**	**Different**	***Shade***
White	-	-	-	-	-	🟨	🟨	
Light	-	-	🟨	-	-	🟨	🟨	
Medium	🟨	🟨	🟨		🟨	-	-	
Dark	🟨	🟨	🟨	🟨	🟨	-	🟨	
Color								

Table 3-9: Tumbling Machine Setup Matrix

		Woolen						Anti Piling						*Pile Type*
		Light		Medium		Dark		Light		Medium		Dark		*Color*
		Same	Different	Same	Different	Same	Different	Same	Different	Same	Different	Same	Different	*Shade*
Woolen	Light	-	🟨	-	-	-	-	▤	▤	▤	▤	▤	▤	
	Medium	-	🟨	-	🟨	-	-	▤	▤	▤	▤	▤	▤	
	Dark	🟨	🟨	-	-	-	🟨	▤	▤	▤	▤	▤	▤	
Anti Pilling	Light	-	-	-	-	-	-	-	🟨	-	-	-	-	
	Medium	-	-	-	-	-	-	-	🟨	-	🟨	-	-	
	Dark	▤	▤	-	-	-	-	🟨	🟨	-	-	-	🟨	
Pile Type	*Color*													

🟨 Minor Cleaning

▤ Major Cleaning

Table 3-10: Gas Setting Machine Setup Matrix

		Light	Medium	Dark		*Color*
		Any	Any	Same	Different	*Shade*
Light	No	-	-	-	-	
Medium	No	-	-	-	-	
Dark	No	▨	-	-	-	
	Yes	🟨	🟨	-	🟨	
Color	*Migration*					

🟨	Major Cleaning
▨	Minor Cleaning

Table 3-11: Flat Fabric Setting Machine Setup Matrix

			Cotton, Rayon, Nylon						*Fabric Type*
			Red	Dark Pink	Navy without Red	Navy with Red	Black without Red	Black with Red	*Color*
			Yes	Yes	Yes	Yes	Yes	Yes	*Migration*
Cotton Rayon, Nylon	Red	Yes	-	-	-	-	-	-	
	Dark Pink	Yes	-	-	-	-	-	-	
	Navy without Red	Yes	-	-	-	-	-	-	
	Navy with Red	Yes	-	-	-	-	-	-	
	Black without Red	Yes	-	-	-	-	-	-	
	Black with Red	Yes	-	-	-	-	-	-	
Poly, TK	Red	Yes	-	-	-	-	-	-	
	Dark Pink	Yes	-	-	-	-	-	-	
	Navy without Red	Yes	-	-	-	-	-	-	
	Navy with Red	Yes	-	-	-	-	-	-	
	Black without Red	Yes	-	-	-	-	-	-	
	Black with Red	Yes	-	-	-	-	-	-	
CVC, TC	Red	Yes	-	-	-	-	-	-	
	Dark Pink	Yes	-	-	-	-	-	-	
	Navy without Red	Yes	-	-	-	-	-	-	
	Navy with Red	Yes	-	-	-	-	-	-	
	Black without Red	Yes	-	-	-	-	-	-	
	Black with Red	Yes	-	-	-	-	-	-	
Others	Light	No	-	-	-	-	-	-	
	Medium	No	-	-	-	-	-	-	
	Dark	No	-	-	-	-	-	-	
Fabric Type	*Color*	*Migration*							

🟨	Major Cleaning
▥	Minor Cleaning

Table 3-11: Flat Fabric Setting Machine Setup Matrix (continued)

			Poly, TK						*Fabric Type*
			Red	Dark Pink	Navy without Red	Navy with Red	Black without Red	Black with Red	*Color*
			Yes	Yes	Yes	Yes	Yes	Yes	*Migration*
Cotton, Rayon, Nylon	Red	Yes	-	-	-	-	-	-	
	Dark Pink	Yes	-	-	-	-	-	-	
	Navy without Red	Yes	-	-	-	-	-	-	
	Navy with Red	Yes	-	-	-	-	-	-	
	Black without Red	Yes	-	-	-	-	-	-	
	Black with Red	Yes	-	-	-	-	-	-	
Poly, TK	Red	Yes	-	-	🟨	-	🟨	-	
	Dark Pink	Yes	-	-	🟨	-	🟨	-	
	Navy without Red	Yes	-	-	-	-	-	-	
	Navy with Red	Yes	-	-	🟨	-	🟨	-	
	Black without Red	Yes	-	-	-	-	-	-	
	Black with Red	Yes	-	-	🟨	-	🟨	-	
CVC, TC	Red	Yes	-	-	🟨	-	🟨	-	
	Dark Pink	Yes	-	-	🟨	-	🟨	-	
	Navy without Red	Yes	-	-	-	-	-	-	
	Navy with Red	Yes	-	-	🟨	-	🟨	-	
	Black without Red	Yes	-	-	-	-	-	-	
	Black with Red	Yes	-	-	🟨	-	🟨	-	
Others	Light	No	-	-	-	-	-	-	
	Medium	No	-	-	-	-	-	-	
	Dark	No	-	-	-	-	-	-	
Fabric Type	*Color*	*Migration*							

🟨 Major Cleaning

▥ Minor Cleaning

Table 3-11: Flat Fabric Setting Machine Setup Matrix (continued)

			CVC, TC						*Fabric Type*
			Red	Dark Pink	Navy without Red	Navy with Red	Black without Red	Black with Red	*Color*
			Yes	Yes	Yes	Yes	Yes	Yes	*Migration*
Cotton, Rayon, Nylon	Red	Yes	-	-	-	-	-	-	
	Dark Pink	Yes	-	-	-	-	-	-	
	Navy without Red	Yes	-	-	-	-	-	-	
	Navy with Red	Yes	-	-	-	-	-	-	
	Black without Red	Yes	-	-	-	-	-	-	
	Black with Red	Yes	-	-	-	-	-	-	
Poly, TK	Red	Yes	-	-	🟨	-	🟨	-	
	Dark Pink	Yes	-	-	🟨	-	🟨	-	
	Navy without Red	Yes	-	-	-	-	-	-	
	Navy with Red	Yes	-	-	🟨	-	🟨	-	
	Black without Red	Yes	-	-	-	-	-	-	
	Black with Red	Yes	-	-	🟨	-	🟨	-	
CVC, TC	Red	Yes	-	-	🟨	-	🟨	-	
	Dark Pink	Yes	-	-	🟨	-	🟨	-	
	Navy without Red	Yes	-	-	-	-	-	-	
	Navy with Red	Yes	-	-	🟨	-	🟨	-	
	Black without Red	Yes	-	-	-	-	-	-	
	Black with Red	Yes	-	-	🟨	-	🟨	-	
Others	Light	No	-	-	-	-	-	-	
	Medium	No	-	-	-	-	-	-	
	Dark	No	-	-	-	-	-	-	
Fabric Type	*Color*	*Migration*							

🟨	Major Cleaning
▥	Minor Cleaning

Table 3-11: Flat Fabric Setting Machine Setup Matrix (continued)

			Others			***Fabric Type***
			Light	**Medium**	**Dark**	***Color***
			No	**No**	**No**	***Migration***
Cotton, Rayon, Nylon	**Red**	**Yes**	🟨	🟨	🟨	
	Dark Pink	**Yes**	🟨	🟨	🟨	
	Navy without Red	**Yes**	🟨	🟨	🟨	
	Navy with Red	**Yes**	🟨	🟨	🟨	
	Black without Red	**Yes**	🟨	🟨	🟨	
	Black with Red	**Yes**	🟨	🟨	🟨	
Poly, TK	**Red**	**Yes**	🟨	🟨	🟨	
	Dark Pink	**Yes**	🟨	🟨	🟨	
	Navy without Red	**Yes**	🟨	🟨	🟨	
	Navy with Red	**Yes**	🟨	🟨	🟨	
	Black without Red	**Yes**	🟨	🟨	🟨	
	Black with Red	**Yes**	🟨	🟨	🟨	
CVC, TC	**Red**	**Yes**	🟨	🟨	🟨	
	Dark Pink	**Yes**	🟨	🟨	🟨	
	Navy without Red	**Yes**	🟨	🟨	🟨	
	Navy with Red	**Yes**	🟨	🟨	🟨	
	Black without Red	**Yes**	🟨	🟨	🟨	
	Black with Red	**Yes**	🟨	🟨	🟨	
Others	**Light**	**No**	-	-	-	
	Medium	**No**	-	-	-	
	Dark	**No**	▨	-	-	
Fabric Type	***Color***	***Migration***				

🟨	Major Cleaning
▨	Minor Cleaning

Chapter 4
Dyeing and Finishing Scheduling Problem Characteristics

With multiple stages, multiple machines, and unique production, dyeing and finishing has a very complicated machine environment. Two machine shop environments, job shop and flexible job shop (i.e., with parallel machines) characterize the dyeing and finishing scheduling problem addressed in this study. Setup is another important issue in that many of the machines have sequence dependent setups. Instead of dealing with the entire job shop or flexible job shop problem directly, the heuristic developed here decomposes the dyeing and finishing shop into sub problems of single machine problems and parallel machine problems.

Since several scheduling environments are discussed, this chapter starts with a brief background. Then, characterization of dyeing and finishing environment is discussed. Finally, the specific dyeing and finishing scheduling problem studied here is discussed.

4.1 General Scheduling Problem Background

4.1.1 Parallel Machine Scheduling Problem

There are *m* machines in parallel. Each job requires only a single operation, and can be processed on any one of the *m* machines.

4.1.2 Job Shop Scheduling Problem

In the general job shop scheduling problem, jobs are processed on a set of machines without preemption. Each job has its own route (production plan) assigning the machine used for each operation. A job can only be processed on the one machine at a time and a machine can only process one job at a time. If a job has to visit certain machines more than once, it is categorized as a re-circulating job shop problem.

4.1.3 Flexible Job Shop Scheduling Problem

A flexible job shop is a modified version of the job shop scheduling problem by combining the parallel machine environments. The shop may have various operations which include a number of identical parallel machines. Each job route specifies the operations and their sequence. Each job is processed at each operation on only one machine and can be processed in any of the parallel machines without preemption. If a job has to visit certain operations more than once, it is categorized as a re-circulating flexible job shop problem.

4.1.4 Sequence Dependent Setup & Sequence Independent Setup Problem

In a schedule, if the setup time depends on both the next job to be processed and the preceding job, the scheduling problem is considered as sequence dependent setup. However, if the setup time depends only on the next job to be processed, the scheduling problem is considered to be sequence independent. In the case of sequence independent setups, they are simply included in the processing time.

4.1.5 Family Scheduling Problem

The family scheduling problem is a modified version of sequence dependent setup problem. In the family scheduling problem, each job belongs to a given family. A setup only occurs when changing from one family to another.

4.2 Dyeing & Finishing Scheduling Problem

4.2.1 Job Shop and Flexible Job Shop Characteristics

In this study, the dyeing and finishing scheduling problem is characterized as a combination of a job shop and a flexible job shop. A set of jobs is processed in multi-stage operation where an operation may have either single machine or parallel machines. Each job has a specified production route. There are two main methods for specifying the machine. In some operations the machine (machine number) is used to indicate the job route (job shop problem). In other operations, the operation (operation number) is used to indicate the job route where the job can be processed in any of the parallel machines (flexible job shop problem). Thus, the production route of each job can specify either the machine number used, or set of machine numbers to be used for processing.

4.2.2 Sequence Dependent Setup & Sequence Independent Setup

The reason for having two different machine specifying methods is a result of the different setup characteristics. If the operation has sequence dependent setup, the machine used for each job is specified prior to production by an expert production controller. For minimizing setups, he (she) tries to allocate jobs having the same features to the same machine. Machine capacity is also one of his (her) considerations. For operations having sequence independent setups, setup minimization is not an issue. Thus, a job can be processed on any of the parallel machines in the operation. The real reason why dyeing and finishing is a combination of a job shop and a flexible job shop, is simply the two different setups involved in production.

4.2.3 Single Machine Problem & Parallel Machine Problem

As noted, the algorithm developed here decomposes the problem into single machine problems and parallel machine problems. The type of problem at each operation is determined from its setup characteristic. There are three possible types of scheduling problems.

- Single machine with sequence dependent setup
- Single machine with sequence independent setup
- Parallel machine with sequence independent setup

Table 4-1 shows the types of scheduling problem (single machine and parallel machines) decomposed for each operation in dyeing and finishing process.

Table 4-1: Types of Scheduling Problem Decomposed for Each Operation

Single Machine Problem	**Parallel Machine Problem**
Dyeing	Spinning
Drying	Cutting
Raising	Brushing
Shearing	
Tumbling	
Setting	
Compacting	

4.2.4 Re-circulating Scheduling Problem

As shown in figure 1-2, a flat fabric order can visit in a flat fabric setting machine more than once (at most twice). In the case of a job visiting the flat fabric setting machine twice, the first visit is called the pre-setting operation. The second visit is the normal setting operation.

After the pre-setting operation, the setting operation can use low temperature which reduces color migration in the fabric.

4.2.5 Family Grouping

Based on the cleaning setup matrices and other setups defined in chapter 3, the criteria for family grouping in each machine are developed. There is no setup incurred between processing two jobs within a family. However, there is a difference in this dyeing and finishing family grouping from the general family scheduling problem. That is, setup may or may not occur when changing from one family to another. The variety of fabrics, colors and finishing requirements in the orders causes many specific families, but some families can be processed next to another without setup.

Dyeing Machine: Jobs are grouped into a family if they have the same following characteristics.

- Fabric type
- Color
- Color percentage range

Tube Drying, Dryer Machine and Compacting Machine: There are 4 main rules for family grouping in these machines.

- The jobs, which are in the white and the light color group, are grouped into a family.
- The jobs, which are in the medium color group, are grouped into a family.
- The jobs, which are in the dark color group, and do not have thermo-migration, are grouped into a family.

- The jobs, which are in the dark color group, have thermo-migration effect, and are in the same shade color, are grouped into a family.

Raising Machine: Jobs are grouped into a family if they have the same following characteristics.

- Fabric type
- Thickness type
- Color group
- Number of rolls range

Shearing Machine: There are four main rules for family grouping.

- Jobs in the white color group are grouped into a family.
- Jobs in the light color group and the same shade color are grouped into a family.
- Jobs in the medium color group and the same shade color are grouped into a family.
- Jobs in the dark color group and in the same shade color are grouped into a family.

Tumbling Machine: Jobs are grouped into a family if they have the same following criteria

- Tumbling type
- Color group (Note: White and light color group are counted as the same color group.)
- Shade color

Gas Setting Machine: There are 4 main rules for family grouping.

- Jobs in the white and light color group, require the same finished fabric width, and use the same setting temperature, are grouped into a family.
- Jobs in the medium color group, require the same finished fabric width, and use the same setting temperature, are grouped into a family.
- Jobs in the dark color group, have no thermo-migration, require the same finished fabric width, and use the same setting temperature, are grouped into a family.
- Jobs in the dark color group and in the same shade color, have thermo-migration effect, require the same finished fabric width, and use the same setting temperature, are grouped into a family.

Steam Setting Machine: Jobs, which have the same setting temperature, are grouped into a family.

Flat Setting Machine: There are 6 main rules for family grouping.

- Jobs whose current operation is pre-setting are grouped into a family.
- Jobs, whose current operation is setting, are in the light and white color group and the same fabric type, and use the same setting temperature, are grouped into a family.
- Jobs, whose current operation is setting, are in the medium color group and the same fabric type, and use the same setting temperature, are grouped into a family.
- Jobs, whose current operation is setting, are in the dark color group and the same fabric type, have no thermo-migration, and use the same setting temperature, are grouped into a family.

- Jobs, whose current operation is setting, are in the dark color group and the same fabric type, have thermo-migration and red shade color, and use the same setting temperature, are grouped into a family.
- Jobs, whose current operation is setting, are in the dark color group and the same fabric type, have thermo-migration but not red shade color, and use the same setting temperature, are grouped into a family.

4.2.6 No Job Priority Classification & Two-Job Priority Classification

Two cases of job priority classification are studied in this research. The first case is a general one in which all jobs have the same priority, named the "***No Job Priority Classification***" case. In the second case, jobs are classified into two priority levels, high and low. The high priority job is a high priority customer's order. Satisfying the due date of a high priority job supersedes the needs of any low priority job. This problem is named the "***Two-Job Priority Classification***" case.

In the two-job priority classification case, job priority becomes one of family grouping criteria. Only jobs within the same priority can be grouped into a family.

4.2.7 Scheduling Objectives

With two cases of problems (i.e., no job priority and two job priority); the scheduling objectives are developed for each case independently. Furthermore, since the research is in multi-stage production with setup times, the lower setup and idle times are also used for determining the best solution. Lower setup time is given higher significance than lower idle time.

4.2.7.1 No-Job Priority Classification Problem

The scheduling objective is to minimize the $L_{\max}$. If two solutions provide equal $L_{\max}$, the solution with the lower setup and/or idle times is kept as the best solution.

4.2.7.2 Two-Job Priority Classification Problem

In the two job priority case, although the main scheduling objective is to minimize $L_{\max}$ of the high priority jobs, improving the $L_{\max}$ of the low priority jobs is treated differently under two conditions of high priority job lateness; negative (early), or positive (tardy). In the condition that the high priority job $L_{\max}$ is tardy, the main concentration is on minimizing the high priority job $L_{\max}$. Minimizing the low priority $L_{\max}$ becomes the second concentration. However, in the condition that the high priority job $L_{\max}$ is negative, the main concentration is on minimizing the low priority job $L_{\max}$ subject to the high priority job $L_{\max}$ remaining negative. In conclusion, there are two scheduling objectives applied to two different conditions of high priority job $L_{\max}$.

Condition 1: The high priority job $L_{\max}$ is positive (tardy).

Scheduling Objective: Minimize high priority job $L_{\max}$

Subject to: Minimize low priority job $L_{\max}$ (second priority)

Condition 2: The high priority job $L_{\max}$ is negative (early).

Schedule Objective: Minimize low priority job $L_{\max}$

Subject to: High priority jobs remain on time.

If two solutions provide equal high priority job $L_{\max}$, the solution with the lowest low priority $L_{\max}$ is kept. If two solutions are equal in both high and low priority $L_{\max}$, the solution with the lowest setup and/or idle time is kept as the best solution.

As mentioned in section 4.2.3, the job shop problem is decomposed into many sub problems (sub single machine problem and sub parallel machine problem). In the developed algorithm, these scheduling objectives are also applied for solving these sub problems with the expectation that the overall scheduling objective will be achieved. However, since in the sub problems all the jobs are in the queue and ready for scheduling (i.e. no idle time occurs), only the setup time is used for comparing two equal $L_{\max}$ solutions.

4.3 Problem Reduction & Assumptions

- Job allocation is excluded from this study. The production route of each job is provided.
- In the parallel operations, all machines are considered to be identical.

Chapter 5
Scheduling Algorithm

The fundamental structure used for solving the dyeing and finishing scheduling problem is the Virtual Factory and family scheduling. The Virtual Factory is based on an idea of Lawrence and Morton [27], and further developed by Hodgson *et al.* [18] for solving the N-job, M-machine, job shop scheduling problem for minimizing maximum lateness (i.e., the $N/M/L_{\max}$ problem).

Family scheduling is used in scheduling problems with sequence dependent setups. Taner [48] and Schultz *et al.* [43] develop a neighborhood search heuristic which the families are sequenced with *EDD* rule, and jobs are exchanged between families, for minimizing $L_{\max}$. Based on their ideas, a scheduling heuristic is developed and coupled with the Virtual Factory in this research.

This chapter starts by providing the background of the Virtual Factory approach. Then the modified version for solving the dyeing and finishing scheduling problem is presented. Scheduling heuristics developed are discussed, and since there are two cases (no job priority and two-job priority classification), each case is discussed separately.

5.1 Introduction to Virtual Factory

The Virtual Factory is a simulation based scheduling approach. The approach consists of repeatedly simulating the system to be scheduled while simultaneously updating job sequences based on the information from the prior simulation. The process is repeated until the lower bound for $L_{\max}$ is achieved, the updating information stabilizes, or a fixed number of iterations are completed.

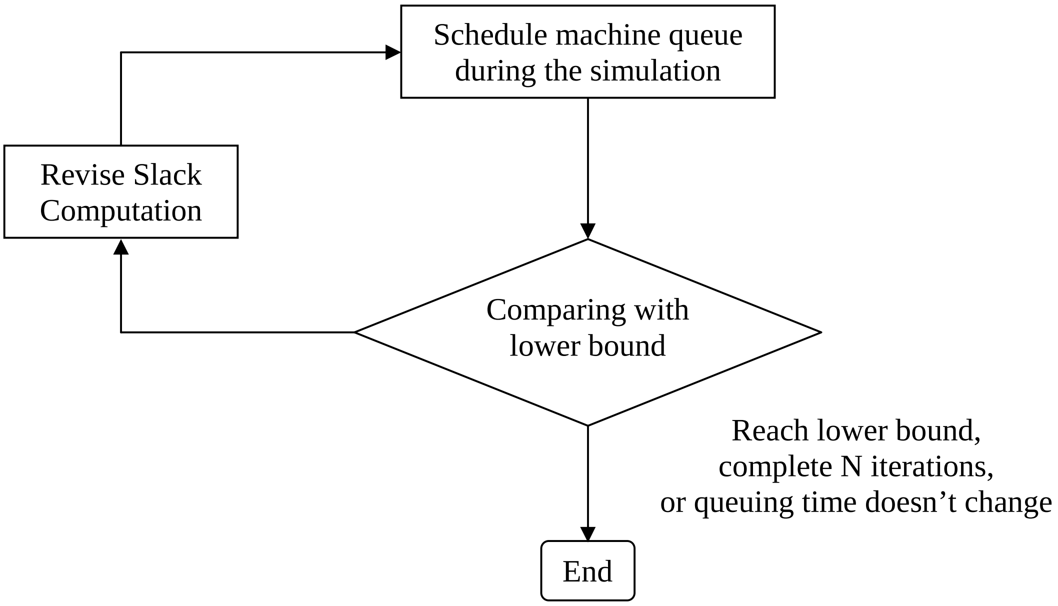

Schedule machine queue during the simulation
Revise Slack Computation
Comparing with lower bound
Reach lower bound, complete N iterations, or queuing time doesn't change
End

Figure 5-1: The Process Flow of the Virtual Factory

There are three main components in the Virtual Factory; lower bound, revised slack and scheduling method. In the Virtual Factory, the job shop problem is broken into many single machine problems. The scheduling method is used to sequence the jobs in the queue for each machine during the simulation. Slack (or effective due date) is used for solving each single machine problem. In the next iteration run, the slack is revised by the queuing information observed from the previous iteration. This slack revising tends to improve the schedule in each iteration. The lower bound is used not only for determining the effectiveness of the solution, but also as a stopping rule for the simulation. Figure 5-1 shows how these three components work together in the Virtual Factory.

5.1.1 A Lower Bound for $L_{\max}$

A well-known and straightforward lower bound is used [9]. Let r_i be the release time of job i, d_i be the due date of job i and $p_{i,j}$ be the processing time of job i on machine j. Assuming infinite capacity on the previous machines, the earliest start time ($ES_{i,m}$) for each job i on machine m is computed as

$$ES_{\mathrm{i,m}} = r_{\mathrm{i}} + \sum_{\mathrm{j \in m^-}} p_{\mathrm{i,j}} \ ,$$

where m^- is the set of all operations prior to machine m on the production route for job i. In like fashion, the latest finish time ($LF_{i,m}$ or slack) for each job i one machine m as

$$LF_{\mathrm{i,m}} = d_{\mathrm{i}} - \sum_{\mathrm{j \in m^+}} p_{\mathrm{i,j}} \ ,$$

where m^+ is the set of all subsequent operations to machine m on the production route for job i.

If $LF_{i,m}$ is interpreted as the effective due date for job i on machine m (d_i^m), and $ES_{i,m}$ as the release time for job i on machine m (r_i^m), each machine can be treated separately as a single machine problem with release times and the objective of minimizing $L_{\max}$. Since this problem is NP-hard, a relaxation using preemption suggested by Baker and Su [3] is used. The earliest due date (*EDD*) rule (with preemption) is known to be optimal for this problem and is used for calculating the lower bound on each machine. Designating the lower bound from machine m as $LB_m(L_{max})$, a lower bound for the job shop problem $LB(L_{max})$ can be obtained as

$$LB(L_{\max}) = \max_{\mathrm{m=1,M}} \{LB_{\mathrm{m}}(L_{\max})\} \, .$$

5.1.2 Revised Slack

Slack (*LF*) is used as an effective due-date, which represents the latest possible time that a job can finish on machine *m*, and still satisfy its final due date:

$$Slack_{\mathrm{i,m}} = d_{\mathrm{i}} - \sum_{\mathrm{j \in m^+}} p_{\mathrm{i,j}} .$$

This version of *slack* does not perform particularly well as a sequencing tool. This is because *slack*, as defined, does not take into account queuing time that may occur as the job is routed over subsequent machines.

A straightforward solution to this weak point is to estimate each job's queuing time at each machine. In subsequent iteration of the simulation, the queuing time observed from the simulation of the previous iteration is used to modify *slack*. Let $q_{i,j}$ be the queuing time of job *i* at machine *j*. *Revised slack* (*Slack*$^{/}$) is defined as

$$Slack^{/}_{\mathrm{i,m}} = d_{\mathrm{i}} - \sum_{\mathrm{j \in m^+}} p_{\mathrm{i,j}} - \sum_{\mathrm{j \in m^{++}}} q_{\mathrm{i,j}} ,$$

where m^{++} is the set of all subsequent operations to machine *m* on the production route except the immediate subsequent operation. The simulation is then rerun using the *revised slack* from the previous iteration.

5.1.3 *EDD* Rule as Scheduling Method

In the simulation, jobs are sequenced in the machine queue in the order of *revised lack*.

5.2 Algorithm for Dyeing and Finishing Scheduling Problem

The algorithm is developed exclusively for two cases (no job priority classification and two-job priority classification). Each case includes both single machine and parallel machine environments, and both sequence dependent and sequence independent setup. The scheduling objective is to minimize $L_{\max}$.

5.2.1 Lower Bound (Estimated Lower Bound)

Although setups are an important part of dyeing and finishing scheduling problem, the lower bound is computed assuming no setups. There are 4 approaches developed for the two cases and the two shop environments; single and parallel machine problem.

- No job priority classification case
    - Single machine problem
    - Parallel machine problem
- Two job priority classification case
    - Single machine problem
    - Parallel machine problem

However, in all 4 situations, the earliest start time ($ES_{i,m}$), the effective due date ($LF_{i,m}$), and a lower bound of the problem $LB(L_{max})$ are computed in the same manner as defined in section 5.1.1.

5.2.1.1 Lower Bound of No Job Priority Classification

Single Machine

In this situation, the scheduling problem for computing the lower bound is the same as the original Virtual Factory mentioned in section 5.1. The earliest due date (*EDD*) rule with preemption is used for determining the lower bound for $L_{\max}$.

Parallel Machines

This situation applies to the operations that have independence setups and therefore are considered as the parallel machine problem. The problem of parallel machines with preemption is computed for the lower bound. Three approaches are discussed for solving the lower bound; network flow, preemptive *EDD* and on-line *LB*.

Network Flow: For determining the optimal, an extension of a network flow approach proposed by Horn [20] and developed by Labetoulle *et a*l. [26] is used. The basic idea of the network flow approach is to construct a series of networks. Each network is tested to see if certain lateness is achievable. By testing lateness values in a systematic fashion over a known range of possibilities, the minimum achievable lateness of all jobs can be found simultaneously, i.e., $L_{\max}$. Unfortunately, this approach takes a lot of computational effort for even small problems.

Preemptive EDD rule: Even though the preemptive *EDD* rule provides the optimal $L_{\max}$ in the single machine problem, Sahni [42] shows that this approach might not provide the optimal solution in the parallel machine problem. In this study, besides the preemption relaxation, another relaxation is added to this rule. That is, a job can be processed on more than one machine at any point of time. Comparing with the network flow approach, this approach requires a lot less computational effort.

On-Line LB: This approach is the minimum $L_{\max}$ approximation for the preemptive parallel machine problem developed by Thoney [49]. The on-line *LB* further relaxes the preemptive problem by allowing the processing of a job on more than one machine during some periods.

In the approach, each job *i* is replaced by p_i unit jobs with the following modified release time and due dated pairs: (r_i, d_i-p_i+1), (r_i+1, d_i-p_i+2),…., (r_i+p_i-1, d_i). These unit jobs are then scheduled by the *EDD* rule, which was shown by Blazewicz [7] to be optimal for the problem of scheduling independent unit length tasks with release times and due dates on identical processors to minimize $L_{\max}$. The example from Thoney[49] of the unit jobs with the pair of modified release time ($r_i^{/}$) and due date ($d_i^{/}$) is shown in table 5-1.

Table 5-1: On-Line Lower Bound Example

Job	$\mathbf{r_i}$	$\mathbf{p_i}$	$\mathbf{d_i}$	**Unit Job ($\mathbf{r_i^{/}}$, $\mathbf{d_i^{/}}$)**
1	0	3	4	(0,2) (1,3) (2,4)
2	0	4	5	(0,2) (1,3) (2,4) (3,5)
3	3	5	10	(3,6) (4,7) (5,8) (6,9) (7,10)
4	5	4	11	(5,8) (6,9) (7,10) (8,11)
5	7	3	13	(7,11) (8,12) (9,13)

5.2.1.2 Lower Bound (Estimated Lower Bound) of Two-Job Priority Classification

In this problem case, the lower bound of high priority job $L_{\max}$ and low priority job $L_{\max}$ are required. A simple procedure was developed for computing the "estimated" lower bound of this case.

Lower Bound Objectives

The scheduling objective for two-job priority classification presented in section 4.2.7.2 is applied as the lower bound objective, but not including the setup comparison between two solutions. This is because it assumes no setup occurred for computing the lower bound. The objectives for the lower bound are as followed.

Condition 1: The high priority job $L_{\max}$ is positive (tardy).

Lower bound Objective: Minimize high priority job $L_{\max}$

Subject to: Minimize low priority job $L_{\max}$ (second priority)

Condition 2: The high priority job $L_{\max}$ is negative (early).

Lower bound Objective: Minimize low priority job $L_{\max}$

Subject to: High priority jobs remain on time.

Procedure for Computing Lower Bound of Two-Job Priority Classification

1. If (Earliest release time of low priority job < Earliest release time of high priority job)

   Schedule the low priority jobs until the earliest release time of high priority job can be processed.

2. Schedule all the high priority jobs.

3. Insert the low priority jobs between two high priority jobs, if it satisfies the following conditions

   Condition 1: At least one of the high priority jobs after the inserted position is tardy. The low priority job can be inserted, if it doesn't make the tardy high priority job tardier.

   This condition means that the low job can be inserted between two jobs, if there is a gap between two jobs and the processing time of the low priority job is less than or equal to the gap range.

Condition 2: All the high priority jobs after the inserted position are early or on time. The low priority job can be inserted, if it does not make any of the on time high priority jobs tardy.

This condition means that the low job can be inserted between two jobs either have gap or have no gap. Furthermore, it can push the jobs after the inserted position backward if it does not make any on time high priority job tardy.

4. Schedule all the remaining low priority jobs by starting sequence them after the last high priority job.

Scheduling Approach for Lower Bound

The approach used for scheduling in steps 1, 2 and 4 of the procedure are the same as computing the lower bound in the no job priority classification case. Again, the choice of approach depends on the shop environment (single or parallel machine). If it is single machine, the preemptive *EDD* rule is applied. If it is parallel machine, the preemptive *EDD* rule with a job being able to process in more than one machine relaxation, or on-line *LB* is applied.

High Priority $L_{\max}$ Lower Bound and Low Priority $L_{\max}$ Lower Bound

The lower bound of high priority job $LB(HL_{\max})$ and low priority job $LB(LL_{\max})$ can be obtained by comparing the high priority lower bound in M machines $LB_m(HL_{\max})$ and the low priority lower bound in M machine $LB_m(LL_{\max})$, respectively.

$$LB(HL_{\max}) = \max_{\mathrm{m=1,M}}\{LB_{\mathrm{m}}(HL_{\max})\}$$

$$LB(LL_{\max}) = \max_{\mathrm{m=1,M}}\{LB_{\mathrm{m}}(LL_{\max})\}$$

An “Estimated” Lower Bound is Used in the Problem

For the no job priority case, the preemptive *EDD* rule provides a “true” lower bound for sigle machine problem. But for the parallel machine problem, there are two possible lower bounds, the preemptive *EDD* rule with a job being able to process in more than one machine, and the on-line *LB*. In the dyeing and finishing process, the parallel machine problem does exist, but those machines have very short processing times. Since the computational time of preemptive *EDD* rule is faster than the on-line LB, and provides bounds that are very close in industrial sized problems, the preemptive *EDD* rule is used. However using *EDD* rule in the parallel machine may not give the true lower bound.

For the two-priority case, a heuristic is developed for estimating the lower bound. Obviously it is an “estimated” lower bound. An error may come from allowing moving the on time priority job backward for inserting the low priority job.

Therefore, in this study the “estimated” lower bound is allowed to use in the virtual factory and in evaluating the scheduling solutions.

5.2.2 Revised Slack with Sequence Dependence

The queuing time observed from the simulation in the previous iteration is used as the estimate queuing time for revising the slack applied in subsequent iterations. Since production requires setup time, now the queuing time of each job in each machine results from both processing time and setup time. In this case two methods for revising slack are applied.

Method 1

The method used for revising the slack in the original Virtual Factory explained in section 5.1.2 is used directly for this problem.

$$Slack^{/}_{\mathrm{i,m}} = d_{\mathrm{i}} - \sum_{\mathrm{j \in m^{+}}} p_{\mathrm{i,j}} - \sum_{\mathrm{j \in m^{++}}} q_{\mathrm{i,j}}$$

Method 2

Slack represents the latest possible time that a job can finish on machine *m*, and still satisfy its final due-date. In method 1, it is assumed that job *i* can jump directly to process in the next machine without waiting in the queue. However, due to the setup requirement, a job may not be able to be processed immediately.

In method 2, the setup time required if job *i* has to be processed immediately in the subsequent operation (m^{+}) to machine m is estimated by using the setup time required before processing job *i* in operation (m^{+}) observed from the previous iteration, and it can be zero. In method 2, the slack is revised as

$$Slack^{/}_{\mathrm{i,m}} = d_{\mathrm{i}} - \sum_{\mathrm{j \varepsilon m^{+}}} p_{\mathrm{i,j}} - s_{\mathrm{i,m^{+}}} - \sum_{\mathrm{j \varepsilon m^{++}}} q_{\mathrm{i,j}}$$

where $\mathrm{s}_{\mathrm{i,m^{+}}}$ is the setup time required if job *i* is processed immediately on machine m^{+}.

5.2.3 Scheduling Methods

The scheduling methods are used to sequence the jobs in the queue of each operation during the simulation. By scheduling each queue individually, the schedule can be treated as a single machine problem or parallel machine problem. In this research, the scheduling methods are developed for two problem cases (no job priority and two-job priority); two shop environments (single and parallel machines); and two types of setup (sequence dependent and sequence independent). In summary, the scheduling methods are developed for solving the following problems.

- No job priority classification case
    - Single machine problem with sequence dependent setup
    - Single machine problem with sequence independent setup
    - Parallel machine problem with sequence independent setup
- Two-job priority classification case
    - Single machine problem with sequence dependent setup
    - Single machine problem with sequence independent setup
    - Parallel machine problem with sequence independent setup

In dyeing and finishing, there are only 3 operations (out of 13 operations), which have sequence independent setups, and they are not the critical processes. The main concentration in this research is therefore to develop the scheduling methods for sequence dependent setup problems. The simple dispatching rule (*EDD*) is modified and applied as the scheduling method for the sequence independent problem.

5.2.3.1 Scheduling Methods for No Job Priority Classification

5.2.3.1.1 Scheduling Method for Sequence Dependent Setup Problem

Concepts used in the Algorithm

Family Scheduling

Since the jobs are grouped into families, the scheduling method actually sequences the families in the queue instead of scheduling each individual job (however, some families may consist of one job).

More Than One Same Family

In a schedule, it is possible to have one family in various positions. This occurs because the scheduling objective is to minimize the $L_{\max}$, not the setup time. Thus, separating jobs categorized in the same family with different due dates into several families may improve $L_{\max}$.

Job Scheduling within a Family

Jobs within a family are sequenced in due date (revised slack) order.

Inserting and Combining Family into a Position

A family can be put in a position in the schedule by one of two methods; inserting and combining. A family can be inserted in front of or behind another family, or between two families, if they are not in the same family. Otherwise, if they are in the same family, two families can be combined.

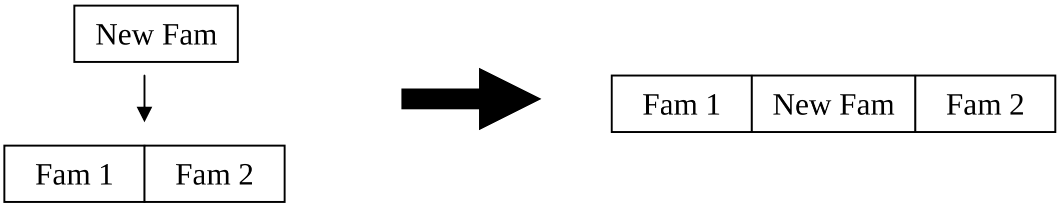


Figure 5-2: Inserting Family Method

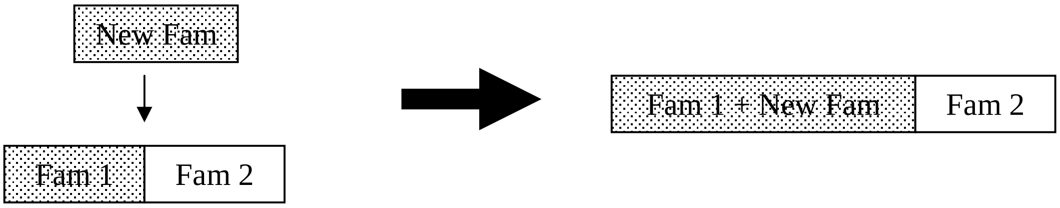


Figure 5-3: Combining Family Method

L_{max} Job and L_{max} Family

The L_{max} job represents the job that has the maximum lateness in the schedule. The L_{max} family is the family that has the L_{max} job as one of its members.

Scheduling Algorithm

The algorithm consists of four scheduling approaches; positioning, switching, L_{max} family splitting, and due after family splitting. The procedure for applying these four approaches is shown in Figure 5-4.

As shown in figure 5-4, the algorithm starts from positioning, switching, L_{max} family splitting, and due after family splitting respectively. In switching and L_{max} family splitting, if the L_{max} family changes after applying the approach, the procedure is started over at the switching step. In due after family splitting, if splitting occurs, the procedure is again started over at the switching step.

Only the positioning is used to put a job into the schedule. The remaining approaches work to improve the schedule and can be applied individually, or in any combination (i.e., it is not necessary to use all 3 approaches).

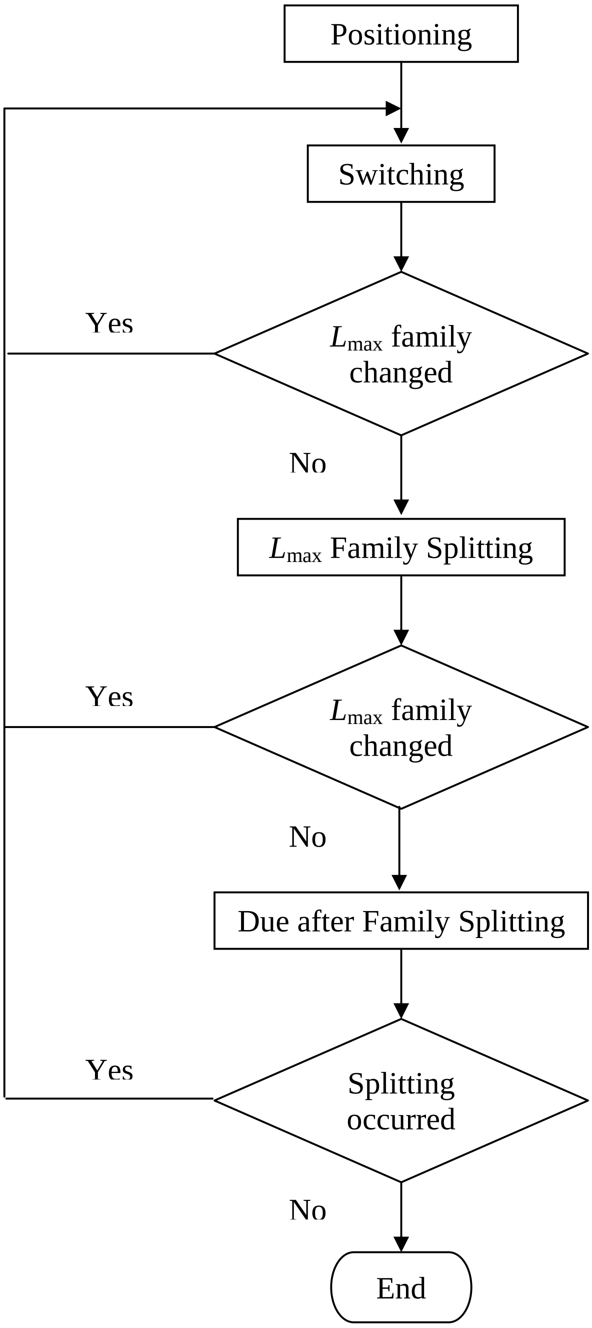


Figure 5-4: Scheduling Algorithm Procedure

Positioning

The approach is to assign a position in the schedule to an incoming job that provides the minimum $L_{\max}$. All positions in the schedule are tested. The new job can be put into the schedule by either inserting it as a new family or combining with an existing same family. As shown in figure 5-5, in the position between two families (or only one family, if the position is the head or tail of the schedule), if one of them is the same family as the incoming job, only the combining family is tested.

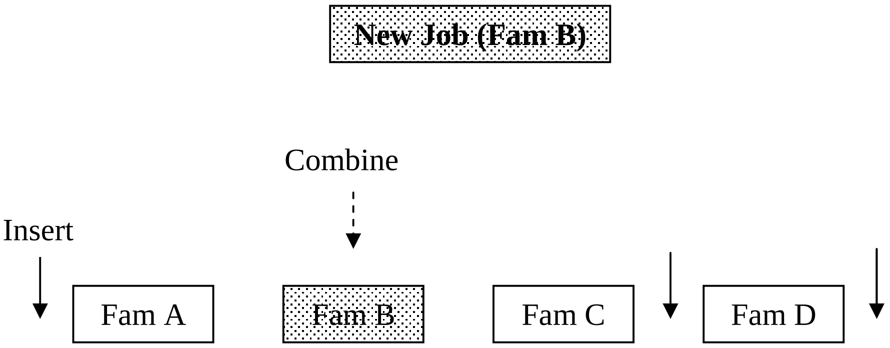


Figure 5-5: Positioning Approach

Switching

The approach is to move the $L_{\max}$ family forward, if it is possible and improve $L_{\max}$. Again the switched family can be inserted between two families or combined with another same family. This step is repeated until there is no improvement in $L_{\max}$. Clearly, this approach cannot work if the $L_{\max}$ family is the first family in the schedule.

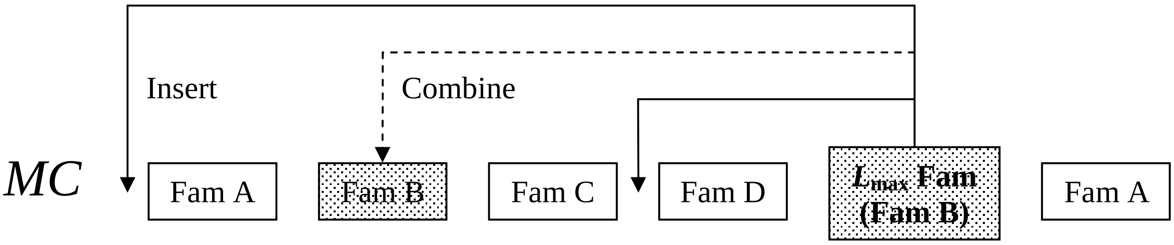
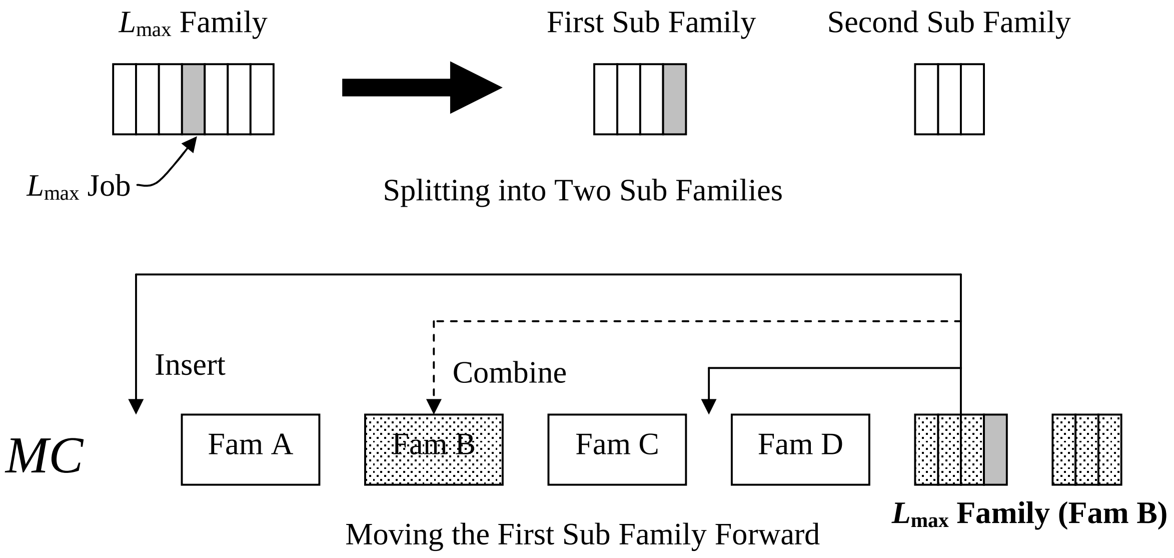


Figure 5-6: Switching Approach

$L_{\max}$ Family Splitting

The approach is to split the $L_{\max}$ family into two sub families and move the sub family including the $L_{\max}$ job forward to improve the schedule $L_{\max}$. In splitting, the first sub-family consists of the jobs up to and including the $L_{\max}$ job. The second sub-family consists of all the jobs after $L_{\max}$ job. The first sub family is moved forward, if moving is possible and so doing improves the schedule. After splitting, if the $L_{\max}$ family is changed, go back to switching approach.

Obviously, if the $L_{\max}$ job is the last job in the $L_{\max}$ family (e.g., the family cannot be split) or the $L_{\max}$ family is the first family in the schedule (e.g., it is not possible to move forward), this approach cannot be applied.



Splitting into Two Sub Families

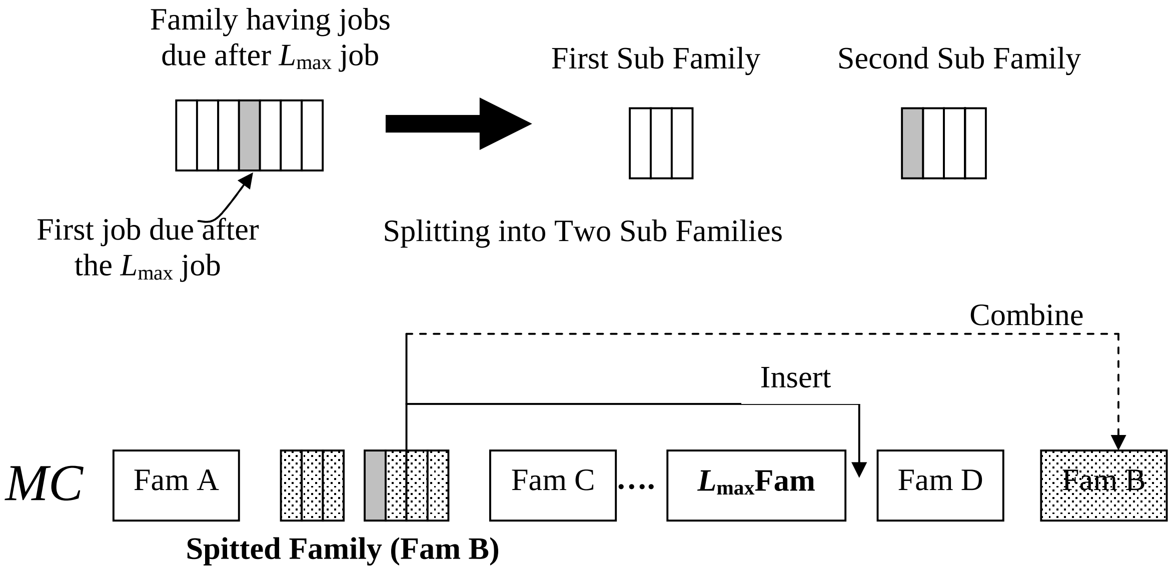
Moving the First Sub Family Forward

Figure 5-7: $L_{\max}$ Family Splitting Approach

Due After Family Splitting

The approach is to split the family that has jobs due after the $L_{\max}$ job but is sequenced before the $L_{\max}$ family into two sub families. The first sub family consists of jobs due before or the same as the $L_{\max}$ job. The second sub group consists of the jobs due after $L_{\max}$ job. The second sub family is moved backward to a position after the $L_{\max}$ group, if so doing improves the schedule $L_{\max}$. If the split occurs, go back to switching.

If the first found job, which has a due date after the $L_{\max}$ job, is the first job in the family, this approach is not applied. This is because putting the $L_{\max}$ family in front of this found family is result in an increase in $L_{\max}$, which is tested in switching approach.


Family having jobs due after $L_{\max}$ job
First Sub Family
Second Sub Family
First job due after the $L_{\max}$ job
Splitting into Two Sub Families
Combine
Insert
MC
Fam A
Fam C
....
$L_{\max}$Fam
Fam D
Fam B
Spitted Family (Fam B)

Moving the Second Sub Family Backward

Figure 5-8: Due After Family Splitting

5.2.3.1.2 Scheduling Method for the Sequence Independent Setup Problem

For minimizing $L_{\max}$, the simple *EDD* rule is applied for the no job priority classification problem with sequence independent setups either in single machine operation or parallel machine operation. For the parallel machine problem, the first job in the schedule is processed by the first available machine.

5.2.3.2 Scheduling Methods for two-Job Priority Classification

5.2.3.2.1 Scheduling Method for Sequence Dependent Setup Problem

Concepts used in the Algorithm

In addition to the concepts (section 5.2.2.1.1) used in the no job priority problem with sequence dependent setup, there are a few more concepts need to be explained before discussing the scheduling method for two-job priority classification problem with sequence dependent setup.

High Priority Zone and Low Priority Zone

In the two-priority classification problem, the schedule is separated into two zones, high and low priority. High priority starts from the first family to the last high priority family in the schedule. Low priority starts from the first family after the high priority zone to the last family of the schedule. The position right after the last high priority family is a special position, which can be included in either the high zone or low zone.

Obviously in this zoning, the low priority zone has only low priority families. As shown in figure 5-9, the high priority zone may include not only high priority families, but also low priority families placed by the switching and splitting steps.

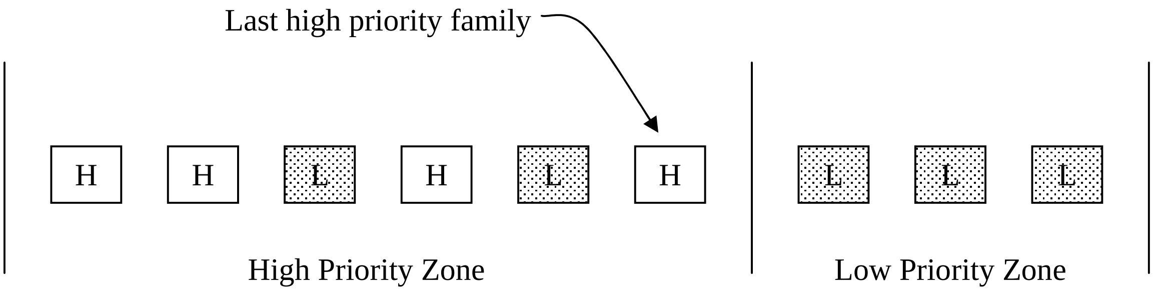


Figure 5-9: High and Low Priority Zone

High $L_{\max}$ Job, High $L_{\max}$ Family, Low $L_{\max}$ Job and Low $L_{\max}$ Family

The high $L_{\max}$ job represents the high priority job that has maximum lateness (compared to other high priority jobs). The high $L_{\max}$ family is the family that contains the high $L_{\max}$ job. Let $HL_{\max}$ be $L_{\max}$ of the high priority jobs.

The low $L_{\max}$ job represents the low priority job that has maximum lateness (compared to other low priority jobs). The low $L_{\max}$ family is the family that contains the low $L_{\max}$ job. Let $LL_{\max}$ be $L_{\max}$ of the low priority jobs.

Scheduling Algorithm

The modified of four approaches; positioning, switching, $L_{\max}$ family splitting, and due after family splitting; discussed in no job priority classification problem are the sequencing tools used in the two-job priority problem. The modification is not only concerned with the two-job priority classification, but also the scheduling objective used (section 4.2.7.2). In the case where $HL_{\max}$ is positive (tardy), the main concern is minimizing $HL_{\max}$. However, if $HL_{\max}$ is negative (early), the main concern is minimizing $LL_{\max}$ while keeping $HL_{\max}$ less than or equal to zero.

The scheduling procedure (figure 5-4) discussed in the no job priority classification is still used for applying all four scheduling approaches. But depending on the priority of the incoming job, only the $L_{\max}$ job that has the same priority as the new job are determined in the procedure, regardless of other low priority jobs. The positions in the schedule, in which the considered family can be placed, are also limited by the priority of the considered family. This will be discussed in detail in the next sections.

Positioning

An incoming job is put in the zone that has the same priority. This means, if the incoming job has high priority, only positions in the high priority zone are considered. If the incoming job has low priority, only positions in the low priority zone are considered.

Positioning Limitation: *Incoming jobs are put in the zone having the same priority*.

With this zone limitation, a high priority job is always put in the front part of the schedule, which supports in minimizing the $HL_{\max}$. It does not support minimizing $LL_{\max}$. However, the main objective is to minimize $HL_{\max}$, and $LL_{\max}$ may be further improved by switching and splitting.

Switching

Depending on the priority of the incoming job, only the same priority $L_{\max}$ family is considered for switching (e.g., If the incoming job has high priority, only the high $L_{\max}$ family is considered. If the incoming job has low priority, only the low $L_{\max}$ family is considered.). Moving forward into positions within its priority zone, there are no limitations. But if a low $L_{\max}$ family is considered for moving into the high priority zone, it can be moved, only if the high priority jobs behind the inserted position don't become tardy (or increase their tardiness).

Switching Limitation: A low priority L_{max} family is able to move forward to the high priority zone if it doesn't make any high priority job after it tardy (or tardier).

Four versions of switching are developed. The switching limitation above holds for all versions. The four versions are as follows:

1. Either on time or tardy low $L_{\max}$ families can move into the high priority zone.

2. Only tardy low $L_{\max}$ families can move into the high priority zone.

3. A low $L_{\max}$ family cannot move past a high $L_{\max}$ family.

4. Only tardy low $L_{\max}$ families can move past a high $L_{\max}$ family, if it doesn't make the on time high priority $L_{\max}$ job tardy (or tardy high priority $L_{\max}$ job tardier).

Since job lateness is computed using revised slack, the on time condition on an upstream machine does not guarantee on time on a downstream machine. With this uncertainty of the job (effective) lateness, it may be necessary to move the low $L_{\max}$ job forward even it is (effectively) on time. For the same reason there is a version such that the low $L_{\max}$ family is not allowed to move past an on-time high $L_{\max}$ family. The effect on $HL_{\max}$ and the performance of these versions is discussed in chapter 6.

$L_{\max}$ Family Splitting

The priority $L_{\max}$ family considered is the one that has the same priority as the incoming job. Family spitting uses the same criterion as the no job priority classification problem. Since in this approach the first sub family (i.e. family including the $L_{\max}$ job) is moved forward similar to switching, the limitations and modifications discussed in switching are applied here.

Due After Family Splitting

The family spitting criterion used is the same as the due after family splitting in no job priority. The priority $L_{\max}$ family considered is the one that has the same priority as the as the incoming job. There are two types of limitations applied to this approach. The first limitation has to do with the criteria for selecting the family to split. If a high $L_{\max}$ family is being considered, any priority family sequenced in front of it can be split. If a low $L_{\max}$ family is being considered, only a low priority family sequenced in front of it can be split. The second limitation has to do with moving a split high priority family backward. The split high priority family cannot move backward past the high priority zone. In both limitations, the approach tries to keep high priority jobs in the front part of the schedule to support $HL_{\max}$ minimization.

Due After Family Splitting:

> *If a low L_{max} family is being considered, only low priority families can be used.*
>
> *A split high priority family cannot move backward past a high priority zone.*

5.2.3.2.2 Scheduling Method for Sequence Independent Setup Problem

For solving this problem, a simple two-step heuristic is developed.

Step 1: The *EDD* rule and priority zone concept are combined together to put a job into a schedule. That is, an incoming job is put into the same priority zone as its priority in due date order.

Step 2: This step works like an improvement step. There are two conditions for doing this step, which depend on the priority of the incoming job.

Condition 1: This step is executed when the incoming job has low priority. If there is a set of on time high priority jobs at the end of the high priority zone, a tardy low job can be moved forward to insert between them if it does not make one of them tardy. This works like switching. Instead of moving the low L_{max} job, the earliest due date tardy low priority job, whose processing time allows insertion, is moved.

Condition 2: This step is executed when the incoming job has high priority. For a tardy high priority job, if there are the low priority jobs sequenced before it, the low priority job(s) is moved backward into the low priority zone. Then condition 1 is checked for the low priority job placed in the low priority zone. This step is repeated until the high priority job is on time or no low priority job is sequenced before it.

5.2.4 Additional Improvement Procedure for Sequence Dependent Setups

The scheduling procedure discussed above is only executed when there is an incoming job in the queue. However, with the setup constraint and a scheduling objective that focuses on only one job (i.e., the L_{max} job), even one job in or out the queue may provide the new improvement opportunities. Thus, an additional simple improvement procedure is provided.

The improvement procedure is to apply switching, L_{max} family splitting, and due after family splitting to the schedule one more time, after a job is left the queue to be processed on a machine. Thus, the queue is scheduled two times, first when a new job comes to the queue, and second when a job leaves the queue to be processed. This may improve the schedule in case the L_{max} job is the job that is put into the machine and there is no incoming job arriving while it is being processed.

Chapter 6
Experimentation and Analysis

The scheduling algorithm developed in this study is tested, and its characteristics and effectiveness in solving the dyeing and finishing scheduling problem are analyzed. The characteristics of actual production, which include the production plan, machine setup types (sequence dependent and independent) and times, and processing times, are applied in problem generation.

The experimental design includes two cases of problems (no job priority classification and two-job priority classification) with various scenarios of the number of jobs and due date ranges, and the number of machines. Several versions of the algorithm are studied in the experimentations.

The model used in the Virtual Factory is explained first. The problem generation and the experimental design used for testing the algorithm are presented. In the experimentation, each version of the algorithm is discussed. The results and analysis of each problem case are presented separately.

As noted in section 5.2.1, the lower bound used is the "estimated" lower bound. Thus the "lower bound" used in this chapter is the "estimated" lower bound. In cases where the pattern of results is consistent across problems whose factors (i.e., job-priority, number of machines) are the same, some of the results obtained are not shown due to redundancy.

6.1 Dyeing and Finishing Process Model in the Virtual Factory

The Virtual Factory mechanism drives the algorithm. The model must mimic the manufacturing process properly. Thus, for solving the dyeing and finishing problem, developing the model that represents the manufacturing process is critical.

Compared to the general job shop, there are two additional features in dyeing and finishing. The first feature is that there may be parallel machines for certain parts of the process. For the parallel machines, the first job in the queue is processed on the first available machine. The second feature is that some machines may have sequence dependent setups. Operations with sequence independent setups are treated as having no setup because the setup is simply included in the processing time. Operations with sequence dependent setups are treated as a single of machine activities. The model is assigned these two features for each operation based on the case plant as shown in table 3-1 and 4-1 and fixed for the entire experimentation. For parallel machines, the number of machines assigned can be varied, and can have just a single machine. Although the features in each operation are fixed in the model, it is easy to modify pending future changes in the case plant.

For academic purposes, a simplified version of the dyeing and finishing model is developed. The simplification is that the wide diversity of job characteristics in the actual plant has been limited somewhat. This simplifies family grouping, and setup and processing time computing. However, the functions are developed based on the actual structure used currently in the case plant. Thus, the model provides the basic fundamental structure of the actual plant.

6.2 Problem Generation

Problems are generated by varying the following factors.

Production route

For each job, a production route is chosen randomly from 168 possible choices. These 168 routes represent most of the production routes used in the case company. The minimum and maximum number of operations is three and eleven, respectively.

By the number of operations specified in the route, the model categorizes the type of shop environment (single/parallel machines) and the types of setup (sequence dependent/independent) and then applies the appropriate algorithm for a job in an operation.

For routes including the pre-setting operation, the job may visit a flat setting machine twice. The same machine is assigned for processing pre-setting and setting operations for that job.

Job Information

As noted, family grouping, setup and processing times are generated by using the simplified structure from the real case plant. Every job is randomly assigned order, setup and processing information. Table 6-1 shows the information that must be generated for all jobs. Table 6-2 shows the optional information that is assigned as necessary.

Table 6-1: The Required Information for All Jobs

Required Information	
Fabric Type	Number of Rolls
Color Group	Weight per Roll
Color	Length per Roll
Weight	Width

Table 6-2: The Optional Information

Optional Information	
Information	**Condition Required**
Thermo-Migration Effect	In the dark color group
Thickness Type	Require surface finishing
Raising Type	Require raising operation
Shear Type	Require shearing operation
Tumble Type	Require tumble operation
Gas Setting Temperature	Require gas setting operation
Steam Setting Temperature	Require steam setting operation
Flat Fabric Setting Temperature	Require flat fabric setting operation
Compacting Temperature	Require compacting operation

Due Date Range

Due dates are sampled from a uniform distribution between 0 and D, where D is defined as the due date range.

Number of Machines

There are two scenarios for specifying the number of machines at each operation. In the first, each operation has one machine (i.e., the one machine case). In the second, the number of machines at each operation is assigned based on the case plant as shown in table 6-3 (i.e., the multiple machine case).

Table 6-3: Number of Machines Used in the Multiple Machine Case

Machine Type	**No. of Machines**
RW 200 kg.	2
RW 400 kg	2
HT 100 kg	5
HT 200 kg	2
HT 400 kg	2
HT 600 kg	3
HT 800 kg	1
HT 1000 kg	3
Spinning	3
Tube Drying	3
Dryer	1
Cutting	1
Raising	10
Brushing	2
Shearing	2
Tumbling	5
Gas Setting	2
Steam Setting	1
Flat Fabric Setting	3
Compacting	1

Job Priority

In the no job priority problem, all jobs are assigned high priority. In the two-job priority problem, job priority is assigned randomly (high or low).

6.3 Experimental Design

Problems are generated with the combination of no job priority and two job priorities, one machine per operation and multiple machines per operation, three numbers of jobs (100, 200 and 400), and six due date ranges (2000, 5000, 8000, 12000, 16000, 20000) as shown in figure 6-1. In each scenario, 200 problems are generated. The number of iterations in the simulation is fixed at 200, if not stated otherwise.

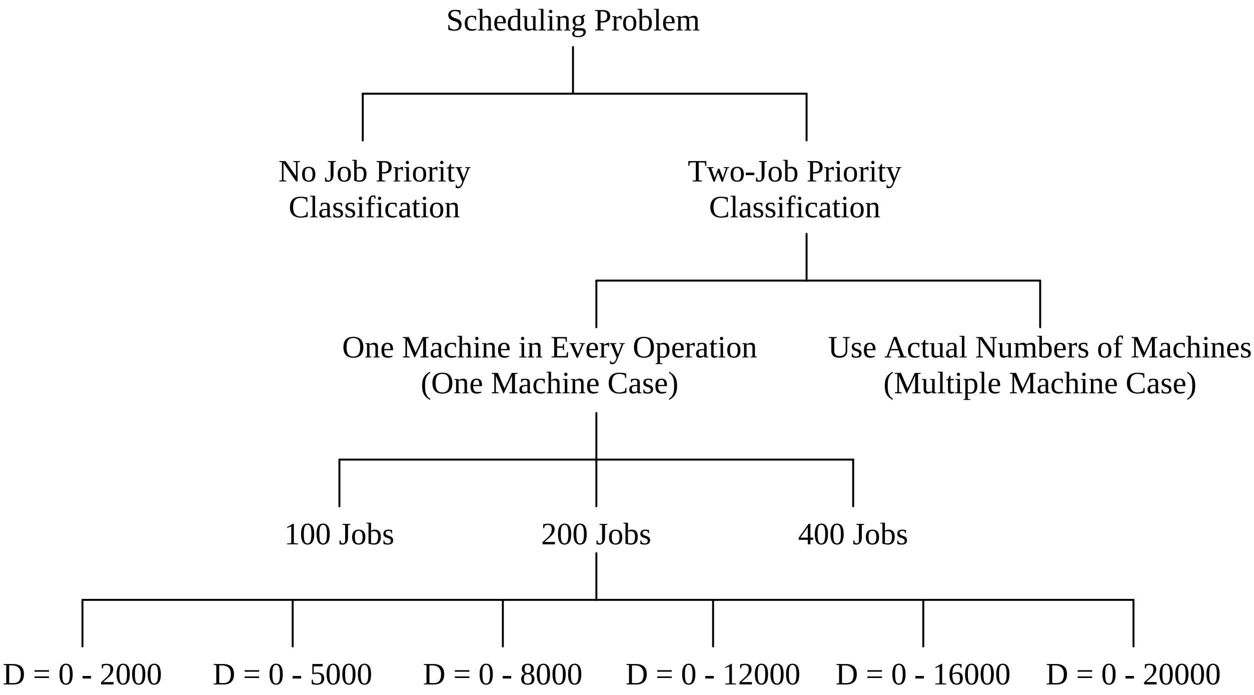

Scheduling Problem
No Job Priority Classification
Two-Job Priority Classification
One Machine in Every Operation (One Machine Case)
Use Actual Numbers of Machines (Multiple Machine Case)
100 Jobs
200 Jobs
400 Jobs
D = 0 - 2000
D = 0 - 5000
D = 0 - 8000
D = 0 - 12000
D = 0 - 16000
D = 0 - 20000

Figure 6-1: Experimental Design

6.4 Experimentations, Results and Analysis

6.4 1 Versions of the Scheduling Algorithm

Three concepts are applied for modifying the scheduling algorithm.

Scheduling When a Job Enters the Queue

For scheduling when a job arrives at a queue, only positioning can be used as an individual scheduling approach. Switching and splitting can be applied with positioning, in any combination. In the experimentation, three combinations of these 4 approaches are tested.

- Positioning
- Positioning and Switching
- Positioning, Switching, $L_{\max}$ Family Splitting, and Due After Family Splitting

Initial Job Arrangement

Initially in the schedule, some machines may have a set of jobs waiting for processing. Since positioning sequences these jobs one by one, changing the job arrangement may change the solution. Three types of job arrangements are applied in the experimentation.

- Jobs are arranged in job number order (randomly).
- Jobs are arranged in due date order (regardless to the job priority)
- High priority jobs are arranged in the first part and low priority jobs are arranged in the second part. In each part, jobs are arranged in due date order.

Re-Scheduling When a Job Leaves the Queue

As noted in section 5.2.4, the algorithm can have an additional step by re-scheduling the queue after a job is put on a machine. The combination of switching and two splitting approaches are applied as follows.

- Switching

- Switching, $L_{\max}$ Family Splitting, and Due After Family Splitting

There are nine versions of the scheduling algorithm for the no job priority problem, and twelve versions for the two-job priority problem tested in the experimentation. The best solution found among all versions is used to determine the algorithm effectiveness. The effectiveness of each version is also determined individually.

6.4.1.1 No Job Priority Classification Problem

The nine versions of the algorithm are as follows. Note that "spitting" is used to refer to both $L_{\max}$ Family Splitting, and Due After Family Splitting.

Table 6-4: Nine Versions of the Algorithm Used in No Job Priority Problem

Method No	**Scheduling approach(s) used when a job enters the queue**	**Initial Job Arrangement**	**Scheduling approach(s) used when a job leaves the queue**
1	Positioning	Job Number	-
2	Positioning	Due Date	-
3	Positioning & Switching	Job number	-
4	Positioning & Switching	Due Date	-
5	Positioning & Switching	Job Number	Switching
6	Positioning & Switching & Splitting	Job Number	-
7	Positioning & Switching & Splitting	Due Date	-
8	Positioning & Switching & Splitting	Job Number	Switching
9	Positioning & Switching & Splitting	Job Number	Switching & Splitting

$L_{\max}$ vs Lower Bound

As shown in the figures 6-2 - 6-5 below, the percentages of problems whose $L_{\max}$ reach the lower bound, or are within the lower bound plus an average processing time, decreases when the due date range increases. The percentages for the one machine case are higher than for the multiple machine case. Furthermore, as shown in figure 6-2 and 6-3 in the one machine case, problems with a higher number of jobs (i.e., 400 jobs) tend to have a higher percentage.

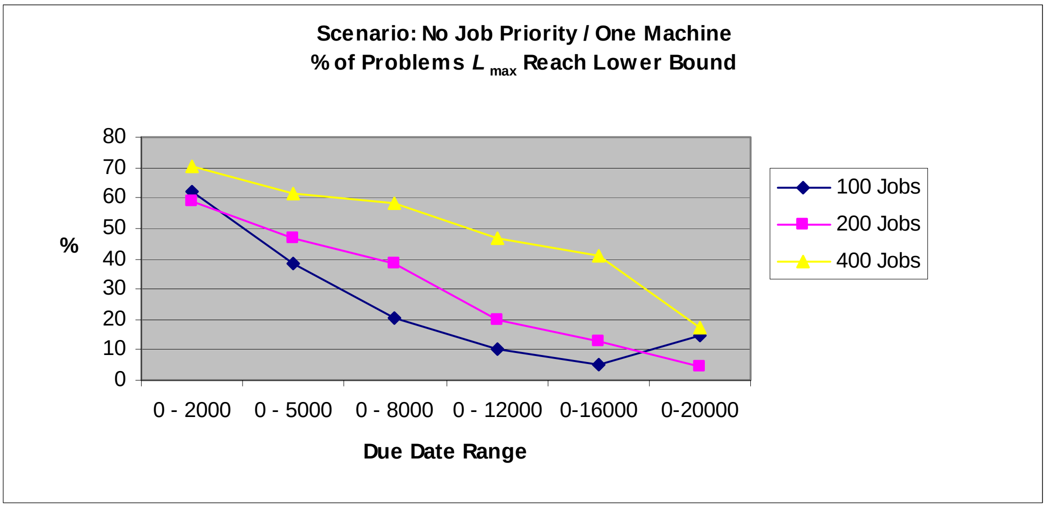


Figure 6-2: Comparing $L_{\max}$ with *LB* in Scenario: No Job Priority / One Machine

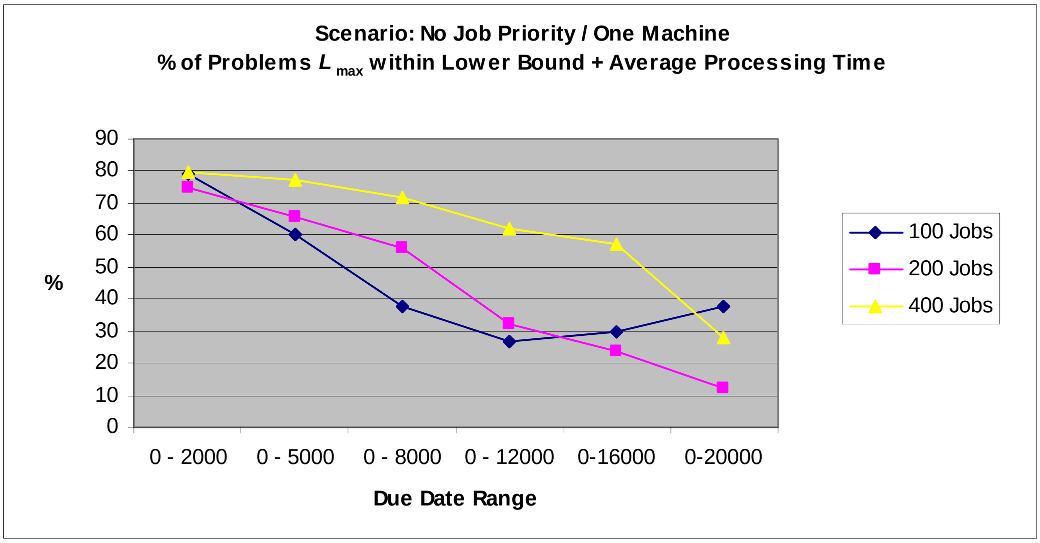


Figure 6-3: Comparing $L_{\max}$ with *LB* plus Average Processing Time in Scenario: No Job Priority / One Machine

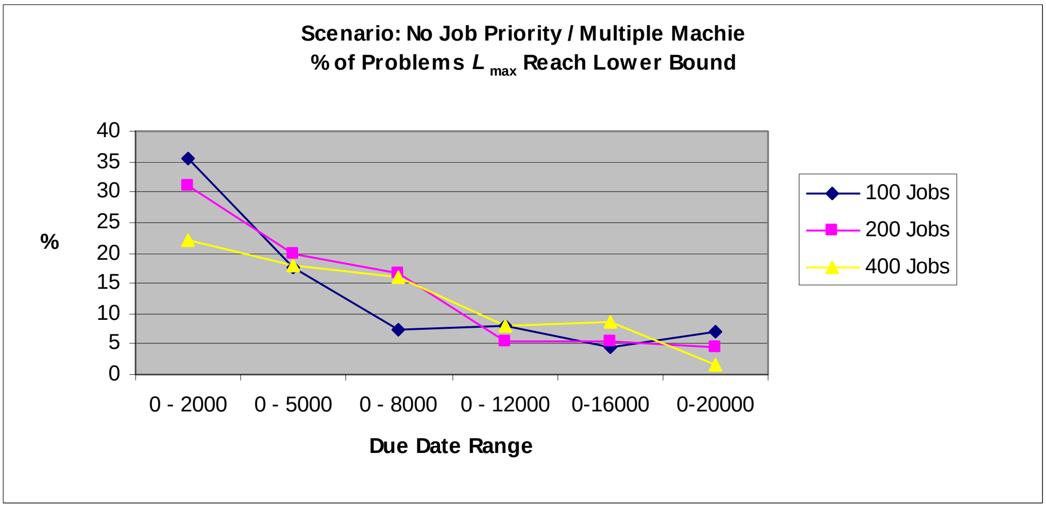


Figure 6-4: Comparing $L_{\max}$ with *LB* in Scenario: No Job Priority / Multiple Machines

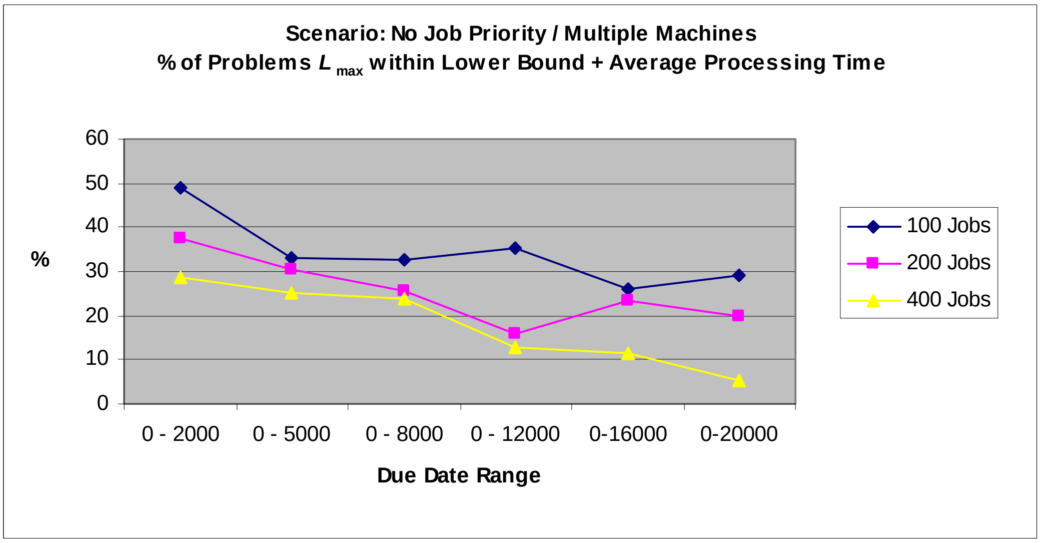


Figure 6-5: Comparing $L_{\max}$ with *LB* plus Average Processing Time
in Scenario: No Job Priority / Multiple Machine

This indicates that the lower bound used here, which does not take account setup times, provides a relatively tight lower bound for the scheduling problems with setup when they have critical production situations. The situations are small due date range and/or low production capacity compared to the number of jobs processed.

Iteration Number of the Best Solution Found

The majority of the best solutions are found in the first ten iterations. The percentage of problems whose best solutions are found in the other iteration ranges is low and approximately equal. The example of this pattern is shown in figure 6-6. Table 6-5 shows the percentage of problems whose best solution is found in the first ten iterations. The scenarios whose percentage are greater than or equal to 50 are highlighted. Obviously, when the number of jobs is high (i.e., 400 jobs), this pattern is strong.

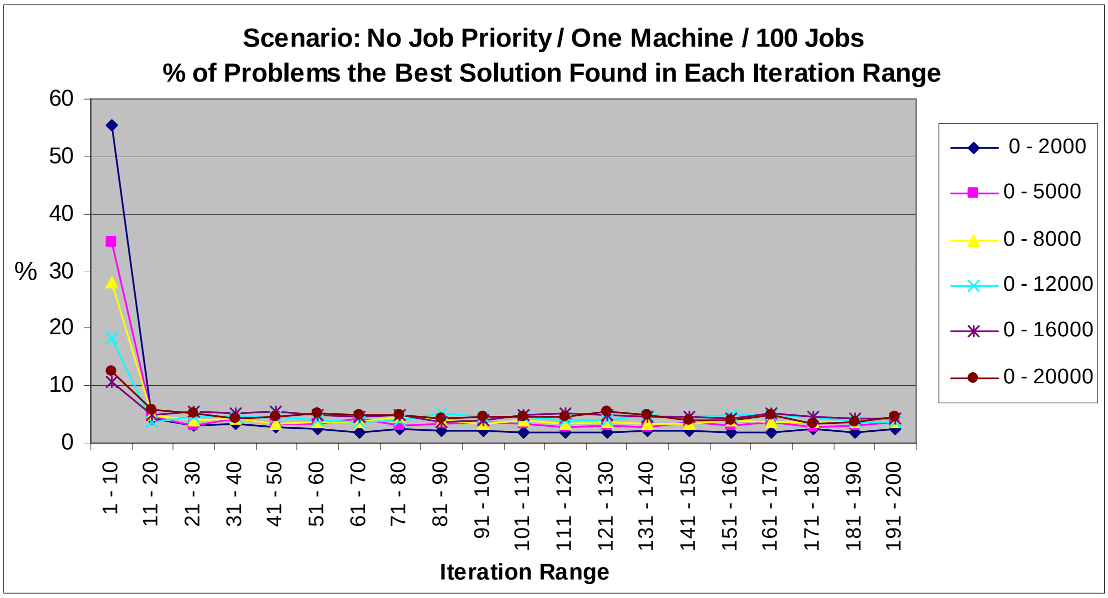


Figure 6-6: The Percentage of Problems the Best Solution Found in Each Iteration Range, No Job Priority Case

Table 6-5: Percentage of Problem the Best Solution Found in the First Ten Iterations, No Job Priority Case

	Number of Jobs	**Due Date Range**					
		0 -2000	**0 – 5000**	**0 - 8000**	**0 - 12000**	**0 - 16000**	**0 - 20000**
One Machine Operation Case	**100**	55.44	35.00	28.11	18.17	10.78	12.61
	200	58.94	51.50	46.61	48.72	39.06	38.44
	400	61.39	54.28	54.61	54.11	50.72	59.56
Multiple Machine Operation Case	**100**	34.67	24.33	24.28	37.72	59.94	66.61
	200	55.89	53.50	48.83	38.50	13.78	12.89
	400	92.17	88.28	86.50	82.72	81.44	78.06

Determining Performace each Individual Scheduling Method

Two types of evaluation are used for determining the performance in each individual method, statistical multiple mean comparison and percentage of problems of each method whose $L_{\max}$ reach the best solutions.

The multiple mean comparison result is shown in appendix A. The observation from the test results are as follows.

- In most problems, all 9 scheduling methods are placed into a single group for all 4 tests, which indicates that there is no statistically significant difference between the means (of $L_{\max}$).
- When the due date ranges are large, the methods may be separated into several groups, which mostly are overlapping.
- In the tests, where the methods are separated into several overlapping groups, there are three interesting observations

    - There is a clear cut between two groups, when the methods that apply only positioning (method 1 and 2) are in one group. These groups are always placed in the bottom of ranking. This indicates that applying positioning alone provides the worst performance.
    - There are several problems which two overlapping groups have most the methods in common. Method 7 is the difference between these two groups. Method 7 is also placed place in the first rank. This may indicate that method 7 provides the best performance.
    - As noted above, method 7 (due date arrangement), not including method 6 (job number arrangement), is in the first rank group. In some problems, only method 3 (job number arrangement), not including method 4 (due date arrangement) is grouped with method 1 and 2 in the lowest rank. This may indicate that due date arrangement performs better than job number arrangement.
- By determining the rank, methods that combine positioning with other approaches (switching and splitting) perform better than using positioning alone. Combining all three approaches performs the best in most problems. Using positioning alone performs the worst in most problems.

By determining the percentage of problems that reach the best solution, it is not clear which scheduling method is the best. The best solution is the lowest $L_{\max}$ found among all methods. For a particular problem, the best solution may be found by several methods (i.e., more than one method may provide the best solution). Figure 6-7 through 6-9 from the multiple machine case are representative, and are used for explaining the observations.

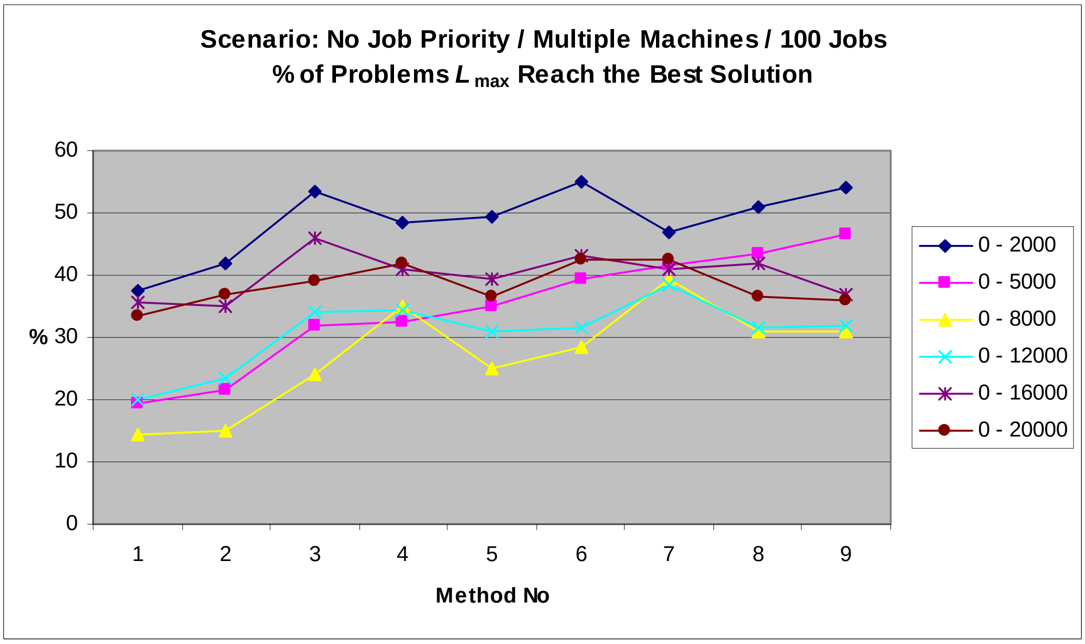


Figure 6-7: Evaluating Each Scheduling Method in No Job Priority Case, 100 Jobs

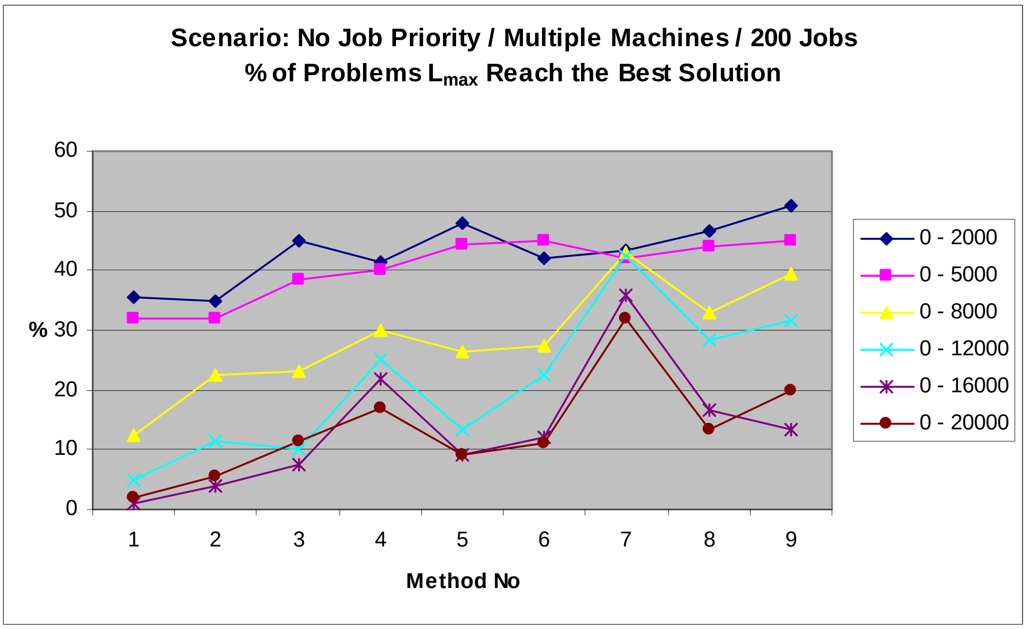


Figure 6-8: Evaluating Each Scheduling Method in No Job Priority Case, 200 Jobs

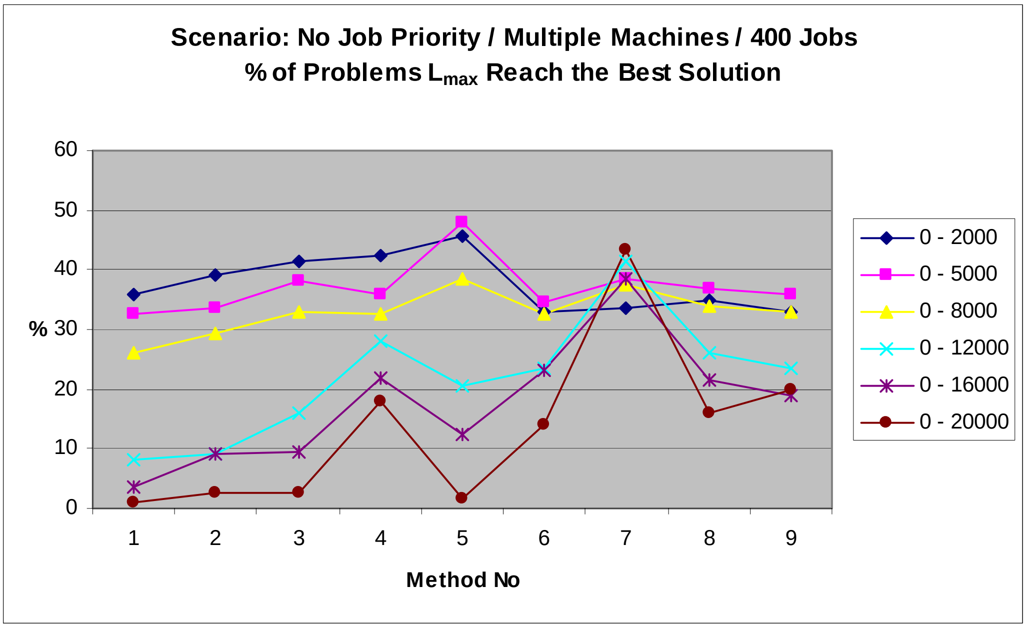


Figure 6-9: Evaluating Each Scheduling Method in No Job Priority Case, 400 Jobs

- The methods that combine positioning with other approaches (switching and splitting) work better than using positioning alone.

- In the large due date ranges and especially when the number of job is high, the combination of all three approaches (positioning, switching, and splitting) works the best.

- As shown in figures 6-8 and 6-9 for 200 and 400 jobs, when the due date ranges are large, method 7 appears to provide the best performance.

- Comparing the two types of initial job arrangement (job number and due date order), the pairs of method 1 and 2, 3 and 4, and 6 and 7 can be used. For the pair of 1 and 2 (positioning used alone), the difference between these two arrangements is not significant. But the pairs 3 and 4 (positioning and switching), and 6 and 7 (positioning, switching and splitting) show significant difference, especially in the pair 6 and 7. When the due date ranges are large, due date order provides better performance (e.g., the figures show the rapidly increase) than the job number order. This pattern is increasingly clear as the number of jobs increases.

- Re-scheduling after a job leaves the queue (methods 5, 8 and 9) does not perform well for improving the schedule overall. Its performance is determined by comparing methods 3 with 5, and method 6 with 8 and 9.

6.4.1.2 Two-Job Priority Classification Problem

The twelve versions of the algorithms used in the two-job priority problem are shown in the following table. Noted here, “H-L Due Date” is the name of the job arrangement case where jobs are put in due date order in their respective priority zones. In all twelve methods, the low priority $L_{\max}$ job can move into the high priority zone whether it is on time or tardy, but it cannot move past the high priority $L_{\max}$ job.

Table 6-6: Twelve Versions of the Algorithm Used in Two - Job Priority Problem

Method No	Scheduling approach(s) used when a job enters the queue	Initial Job Arrangement	Scheduling approach(s) used when a job leaves the queue
1	Positioning	Job Number	-
2	Positioning	Due Date	-
3	Positioning	H-L Due Date	-
4	Positioning & Switching	Job Number	-
5	Positioning & Switching	Due Date	-
6	Positioning & Switching	H-L Due Date	-
7	Positioning & Switching	Job Number	Switching
8	Positioning, Switching & Splitting	Job Number	-
9	Positioning, Switching & Splitting	Due Date	-
10	Positioning, Switching & Splitting	H-L Due Date	-
11	Positioning, Switching & Splitting	Job Number	Switching
12	Positioning, Switching & Splitting	Job Number	Switching & Splitting

_L_max vs Lower Bound

The percentage of problems whose $HL_{\max}$ reach the lower bound of the high priority job decreases as the due date range increases. The percentage decreases sharply at the beginning and then stabilizes in the larger due date ranges. The (large) 400 job problems have a higher percentage than the (small) 100 job problems, when the due date ranges are small. But when due date ranges are large, the percentages of the 100 job problems is higher.

The lower bound also seems to perform well when the due date range is small and the production capacity is low relative to the number of jobs. In figure 6-12 and 6-13, the percentage of problems whose $HL_{\max}$ is within the range of the lower bound plus the average processing times is shown.

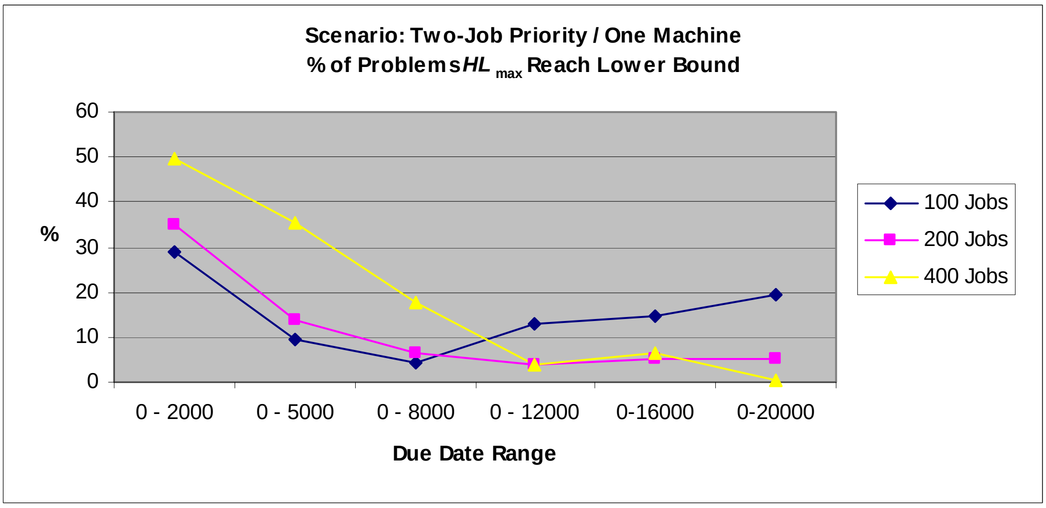


Figure 6-10: Comparing $HL_{\max}$ with *LB* in Scenario: Two-Job Priority / One Machine

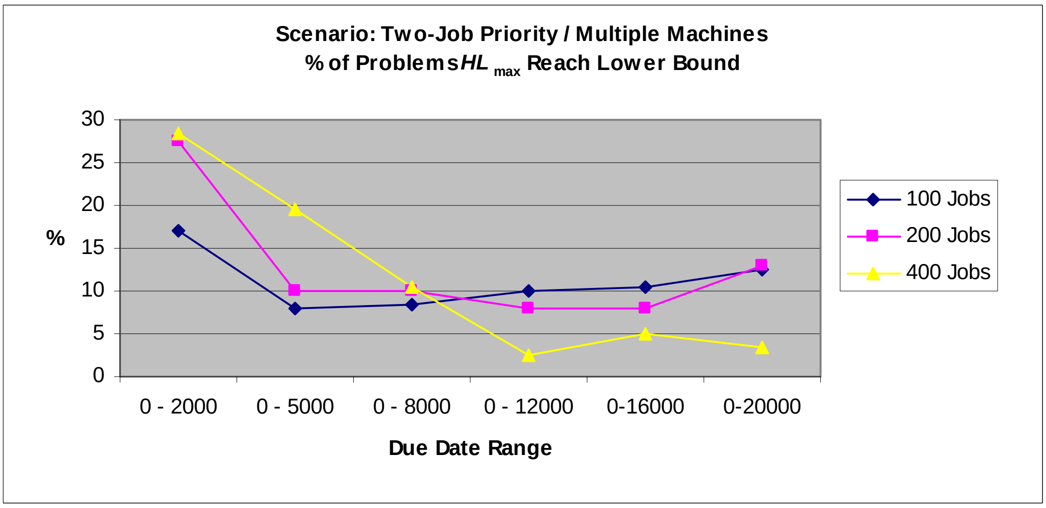


Figure 6-11: Comparing $HL_{\max}$ with *LB* in Scenario: Two-Job Priority/Multiple Machines

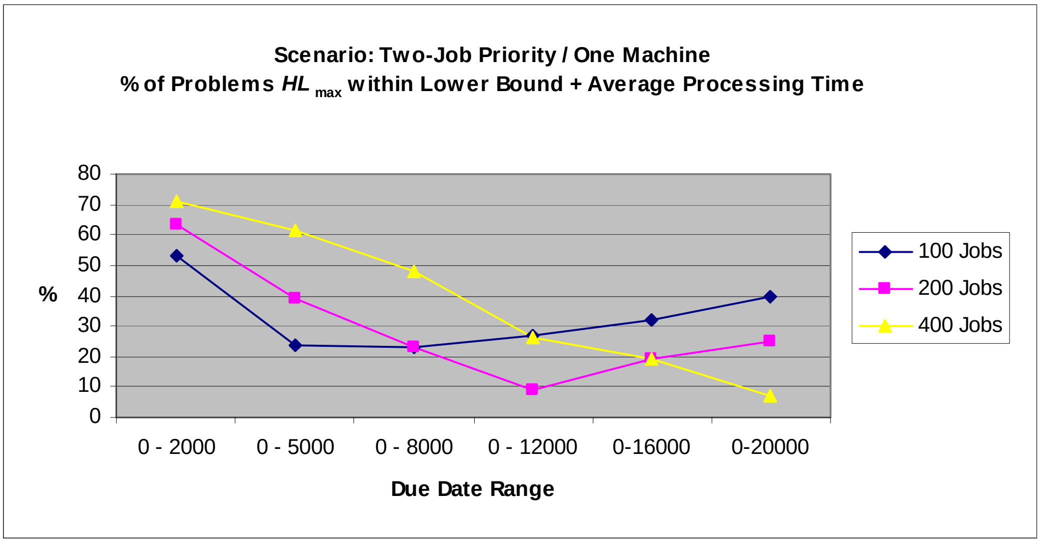


Figure 6-12: Comparing $HL_{\max}$ with LB plus Average Processing Time
in Scenario: No Job Priority / One Machine

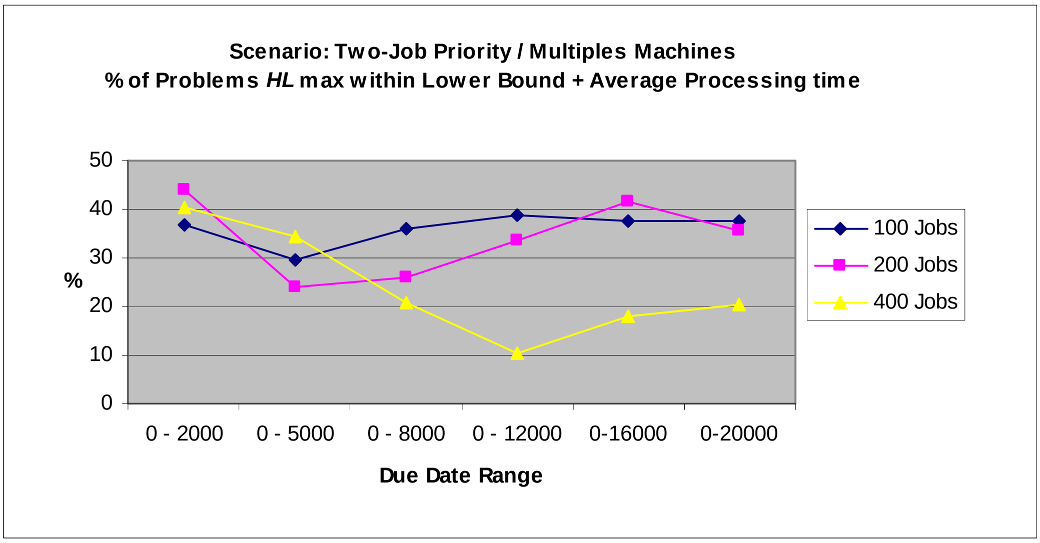


Figure 6-13: Comparing $HL_{\max}$ with LB plus Average Processing Time
in Scenario: No Job Priority / Multiple Machines

In figure 6-14 and 6-15, the percentage of problems whose both $HL_{\max}$ and $LL_{\max}$ reach the lower bounds (e.g. the high priority $L_{\max}$ and low priority $L_{\max}$ lower bound) also decreases as the due date range increases. However, in the large due date ranges, the percentage is very close to zero or zero.

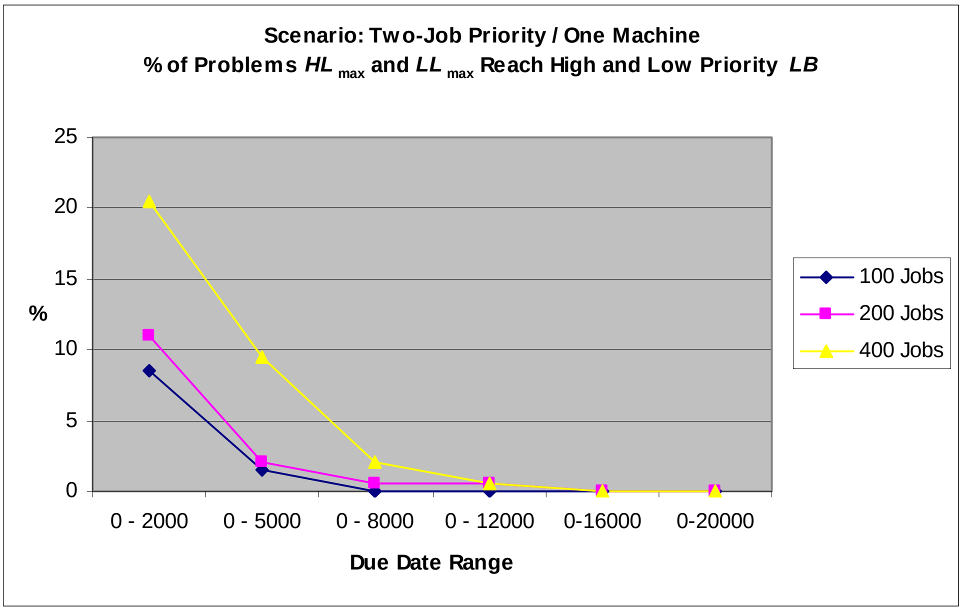


Figure 6-14: Comparing HL_{max} and LL_{max} with *LBs* in Scenario: Two-Job Priority / One Machine

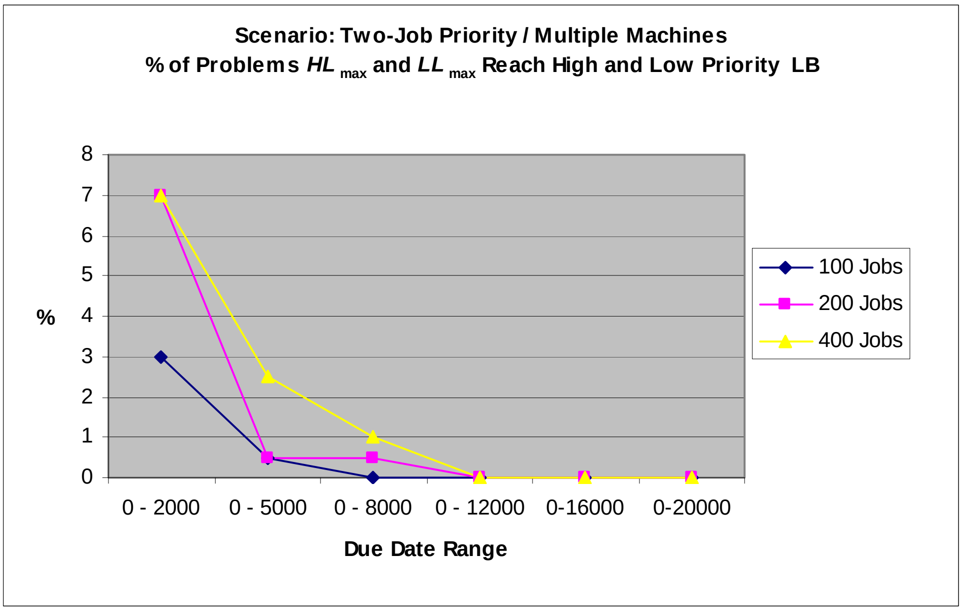


Figure 6-15: Comparing HL_{max} and LL_{max} with *LBs* in Scenario: Two-Job Priority / Multiple Machines

In figure 6-16 and 6-17, the performance of the algorithm is shown by the percentage of problems whose best solutions provides both the lowest $HL_{\max}$ and $LL_{\max}$ among the 12 methods. The percentages are relatively high compared with the percentage both reached lower bound in all cases. The percentage decreases as the due date range increases, but in the 100 and 200 job problems, it bounds back at the larger due date ranges.

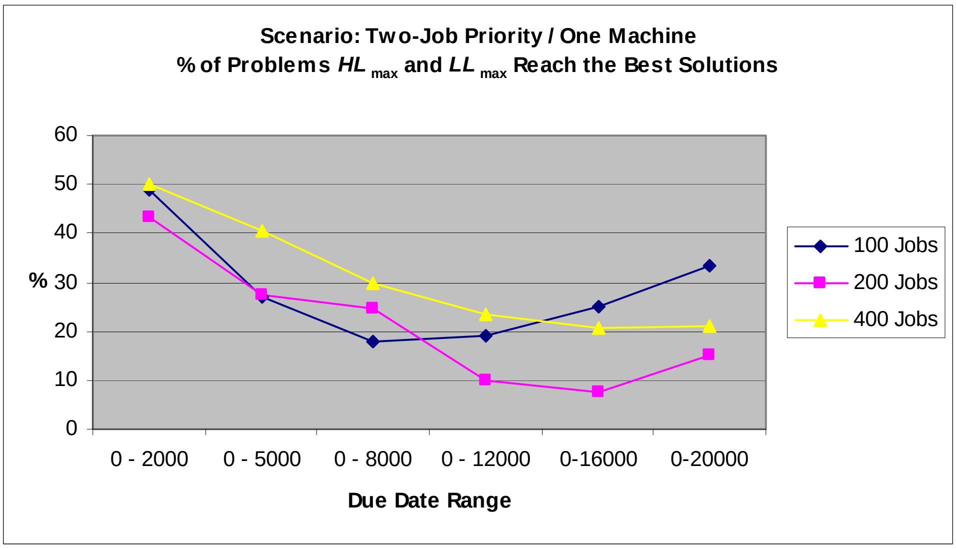


Figure 6-16: Comparing $HL_{\max}$ and $LL_{\max}$ with the Best Solutions in Scenario: Two-Job Priority/One Machine

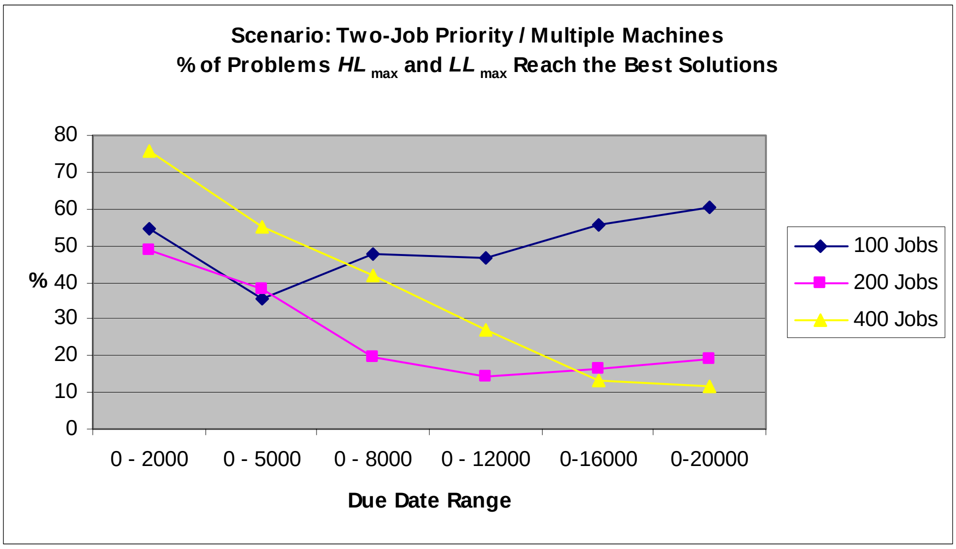


Figure 6-17: Comparing $HL_{\max}$ and $LL_{\max}$ with the Best Solutions in Scenario: Two-Job Priority/Multiple Machines

Iteration Number of the Best Solution Found

Comparing table 6-7 with table 6-5, the percentages found in the first ten iterations of the two-job priority are obviously lower than the no-job priority classification case. As shown in figure 6-18, the percentage of problems whose best solutions are found in the other iteration number ranges is low and approximately equal except the iteration number 11 – 20 and 21 – 30 where the percentages are a little higher than the others.

Table 6-7: Percentage of Problems the Best Solution Found in the First Ten Iterations, Two-Job Priority Case

	Number of Jobs	**Due Date Ranges**					
		0 -2000	**0 – 5000**	**0 - 8000**	**0 - 12000**	**0 - 16000**	**0 - 20000**
One Machine Operation Case	**100**	19.46	13.04	8.79	12.63	19.63	25.00
	200	28.71	25.25	22.79	19.79	10.29	10.25
	400	30.33	29.96	29.75	34.92	34.92	39.25
Multiple Machine Operation Case	**100**	32.96	44.29	57.75	72.04	86.13	88.13
	200	25.75	17.21	10.83	9.13	12.63	17.67
	400	55.08	49.88	49.88	40.67	16.46	14.63

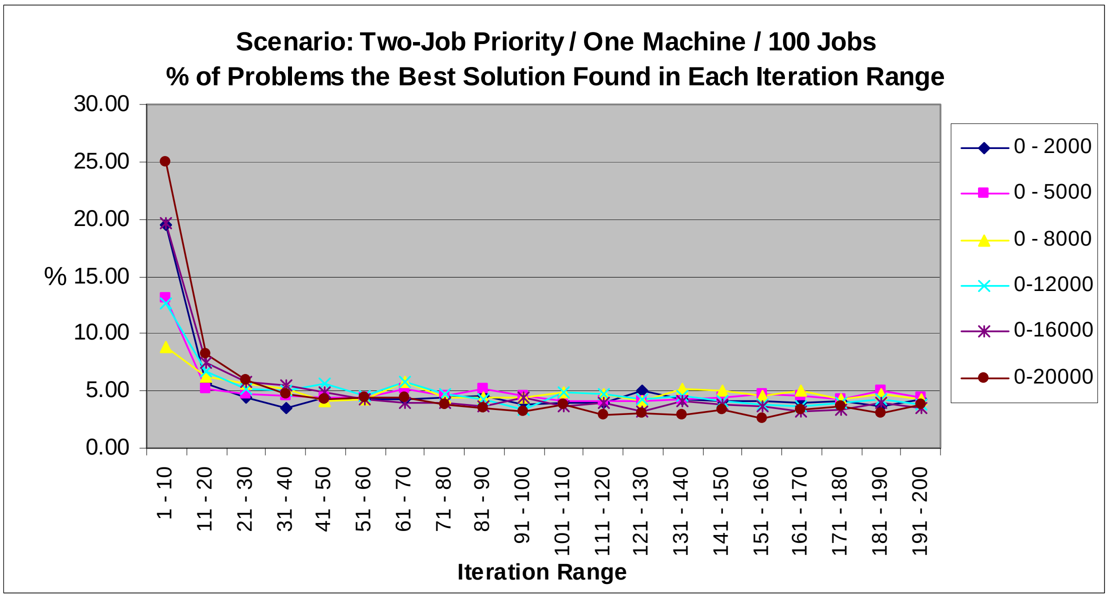


Figure 6-18: The Percentage of Problems the Best Solution Found in Each Iteration Range, Two-Job Priority Case

Determining Performance each Individual Scheduling Method

Again, statistical multiple mean comparison and percentage of problems of each method, whose $L_{\max}$ reach the best solutions, are used for evaluating each method.

From the results of the multiple mean comparison (Appendix A), the same patterns occurred in two-job priority case are quite the same as the no job priority case. Their observations are as follows.

- In most problems, there is no statistical significant difference for the mean of $HL_{\max}$ between these 12 methods (i.e., they are placed in a single group).
- When the number of jobs and due date range is large, the methods may be separated in several groups, which mostly are overlapping.
- Interesting observations, when there are several overlapping groups

    - In several problems, the methods that apply only positioning are placed in one group (in the worst rank). This indicates that applying positioning alone provides the worst performance.
    - When two overlapping groups obtain most methods in common, method 9 and/or 10 are the differences between them. Since method 9 and 10 are in the first rank in most problems, this may indicate that combining all three approaches performs the best.
    - From above discussion, only method 9 and 10 which apply due date related arrangement, are the different methods between two overlapping groups. It does not include method 8, which apply the same scheduling approach but with job number arrangement. This may indicate that due date arrangement performs better than job number arrangement.
- By determining the rank, methods that combine positioning with other approaches (switching and splitting) perform better than using positioning alone. Combining all three approaches performs the best in most problems. Using positioning alone performs the worst in most problems.

There is also no evidence that shows which method performs the best or which method should be eliminated, when the percentage of problem reached the best solution is determined. The best solution is likely to be found in any one(s) of 12 methods. As with the no job priority problems, the same observations are applied in both single and multiple machine cases. Both cases show a similar effect in all observations. The multiple machine problems in figures 6-19 through 6-21 are used to explain the observations.

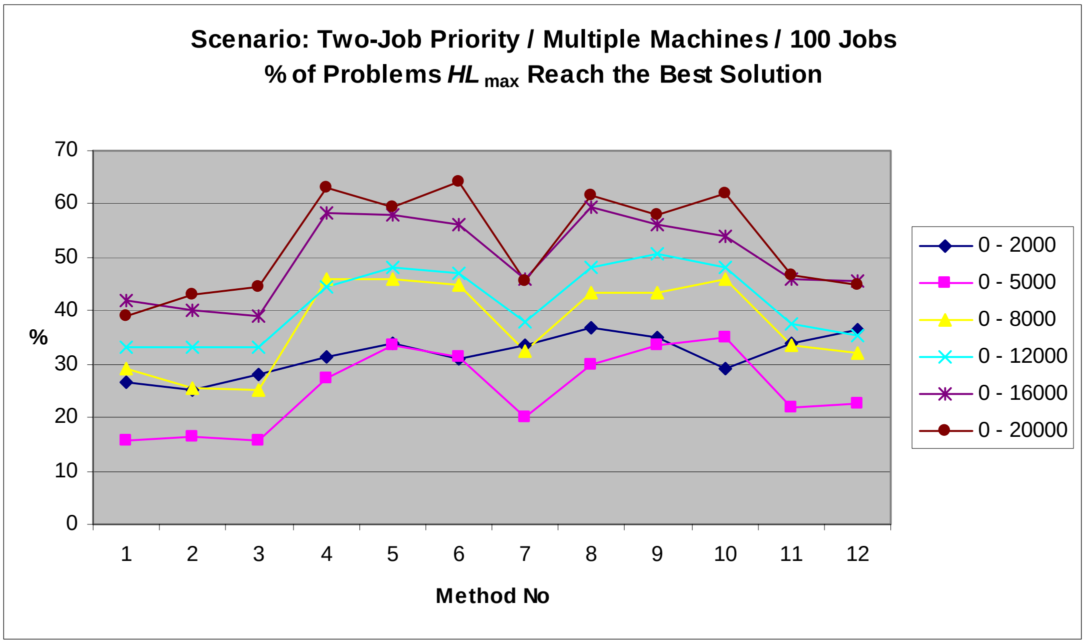


Figure 6-19: Evaluating Each Scheduling Method in Two-Job Priority Case, 100 Jobs

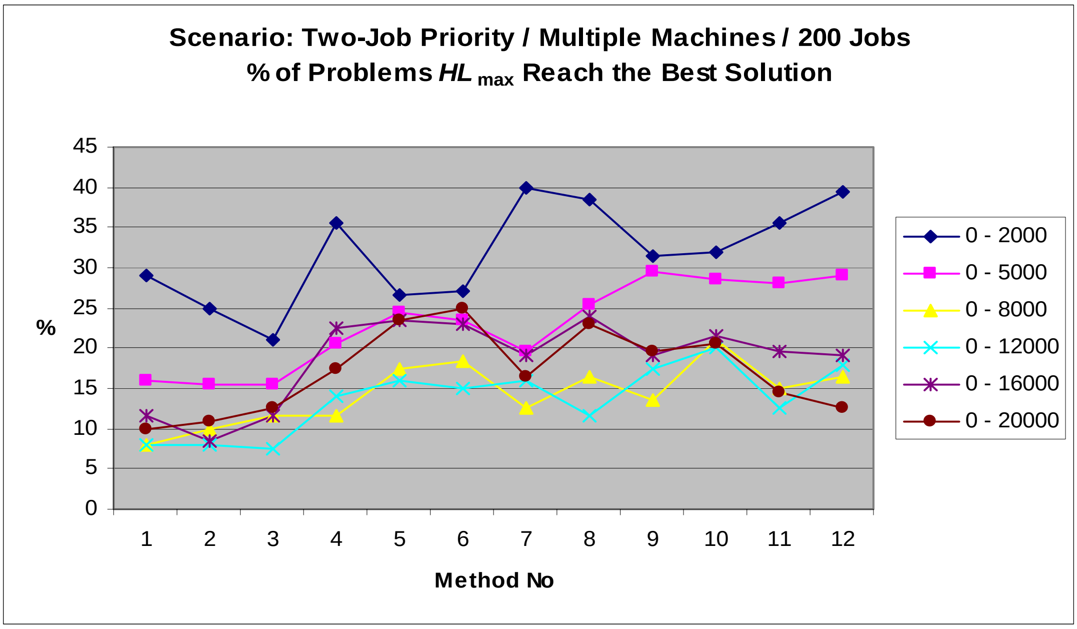


Figure 6-20: Evaluating Each Scheduling Method in Two-Job Priority Case, 200 Jobs

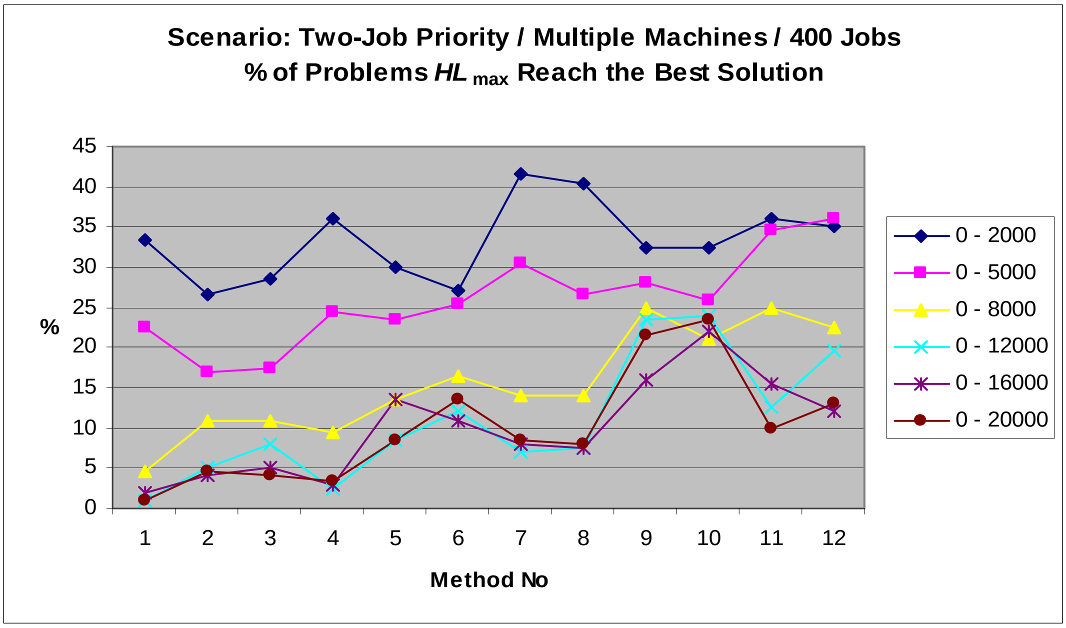


Figure 6-21: Evaluating Each Scheduling Method in Two-Job Priority Case, 400 Jobs

- The combination of positioning with other approaches (switching and splitting) performs better than using the positioning alone.

- When the number of job is low (100 jobs), combining positioning and switching provides the same performance as combining positioning, switching, and splitting as shown in figure 6-19.

- In the higher numbers of jobs (200 and 400 jobs), combination of all three approaches (positioning, switching, and splitting) performs the best, especially when the due date ranges are large.

- There seems to be no significant difference between the three types of job arrangement when positioning is applied alone.

- In the combined approach methods, due date related order (e.g., due date or priority zone due date) provide better performance than job number order when the due date ranges are large. This pattern is becomes clearer as the number of jobs increases.
- When the number of jobs is high (e.g. 400 jobs) and the due date range is large, applying due date related order with the combination of the three approaches works better than with the combination of positioning and switching shown in figure 6-21.
- Again, re-scheduling when a job leaves the queue seems to perform well when the number of job is high. Its performance is determined by comparing methods 4 with 7, and method 8 with 11 and 12.

6.4.2 Moving the low $L_{\max}$ Family into the High Priority Zone

As mentioned in section 5.2.3.2.1, the switching and $L_{\max}$ family splitting approaches can be modified by setting the following rules. The first rule is to limit the tardiness (i.e., on time or tardy) of the low priority $L_{\max}$ family that can pass into the high priority zone. The second rule is to limit the position that the low priority $L_{\max}$ family can go through (i.e., can or cannot pass the high priority $L_{\max}$ family). Three alternatives are developed by applying these two rules. Each alternative is used in the algorithm and its performance tested.

Alternative 1: The low priority $L_{\max}$ family can move into the high priority zone either on time or tardy, but cannot be moved past the high priority $L_{\max}$ family. (Note, this alternative is the one that is used in the algorithm solving all two-job priority problems in the above experimentation.)

Alternative 2: Only the tardy low priority $L_{\max}$ family can move into the high priority zone, but cannot be moved past the high priority $L_{\max}$ family.

Alternative 3: The low priority $L_{\max}$ family can move into the high priority zone either on time or tardy. Only a tardy family can be moved past the high priority $L_{\max}$ family, if it doesn't make the on time high priority $L_{\max}$ job tardy (or tardy high priority $L_{\max}$ job tardier).

The performance of these three Alternatives is determined comparing the high priority job $L_{\max}$. The results from alternative 1 are used as a reference for comparing with the other alternatives. The percentage of problems in which the high priority job $L_{\max}$ decreases, increases or ties with alternative 1 is used as an indicator for determining performance.

Alternative 1 vs Alternative 2

Figure 6-22 shows there are no difference between these two alternatives when the due date range is very small (e.g. 2000). In the higher due date ranges, figure 6-22 shows there are significant differences (e.g., decrease or increase). However, the percentage of problems in which the high priority $L_{\max}$ job for alternative 2 increases and decreases from alternative 1 are close to each other. Thus, there is no evidence of which alternative provides the better solution. It is equally that one provides a better solution than the other.

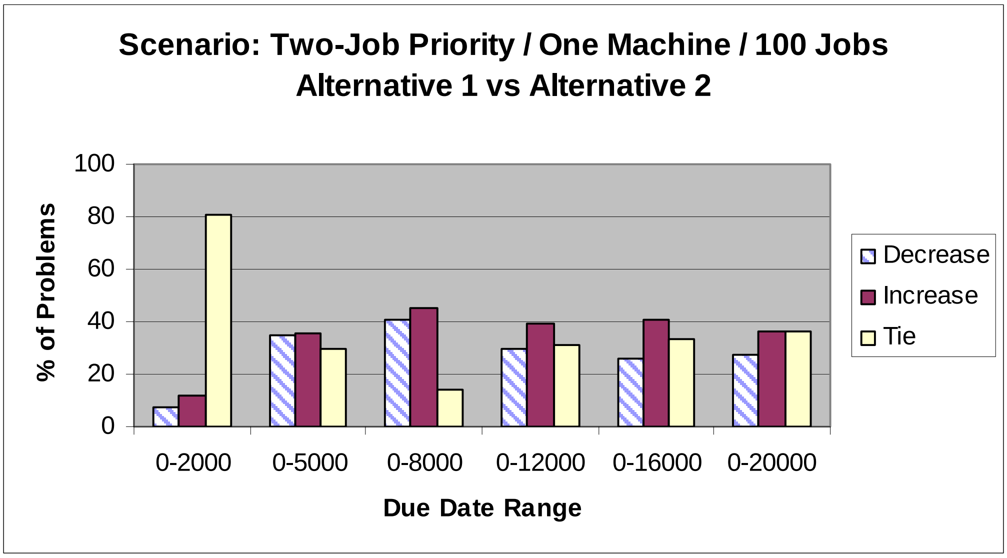


Figure 6-22: Evaluating Two Moving Low Priority L_{max} Family Alternatives, 1 and 2 in Scenario: Two-Job Priority / One Machine / 100 Jobs

Alternative 1 vs Alternative 3

The change in the solution between these two alternatives is not significant when the due date range is small (0 – 2000 or 0 – 5000), especially when the number of jobs is high as shown in figure 6-24. The effect (e.g. the percentage of problems whose solutions are different) of them increase as the due date range is higher. Comparing these two alternatives, the percentages are obviously greater than the comparison between alternative 1 and 2. When the number of jobs is high, the percentage of problems increasing and decreasing are close as shown in figure 6-24. When the number of jobs is low, the percentage of problems whose high priority L_{max} decreases is significantly higher than those increasing, as shown in figure 6-23. Since Alternative 3 allows the low priority job to move past the high priority L_{max} job, the change in the high priority L_{max} is expected to be increasing not decreasing. However, with this allowance the scheduling heuristic can search for a better solution.

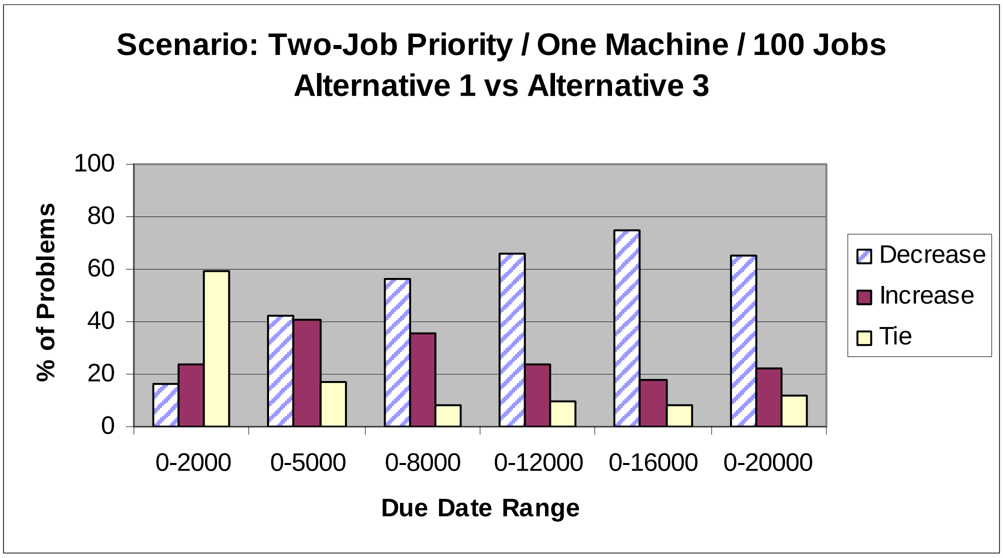


Figure 6-23: Evaluating Two Moving Low Priority $L_{\max}$ Family Alternatives, 1 and 3 in Scenario: Two-Job Priority / One Machine / 100 Jobs

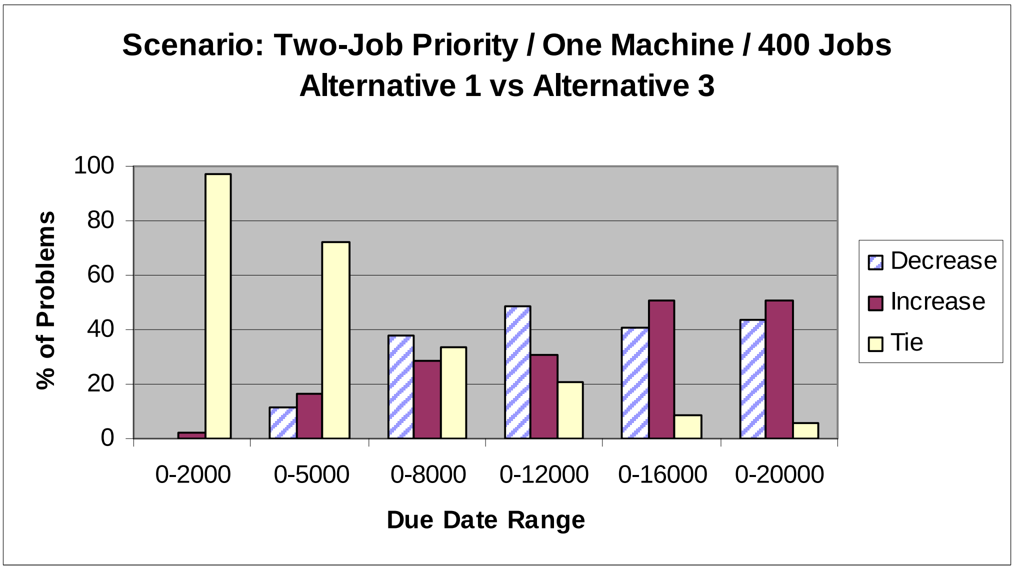


Figure 6-24: Evaluating Two Moving Low Priority $L_{\max}$ Family Alternatives, 1 and 3 in Scenario: Two-Job Priority / One Machine / 400 Jobs

As noted in section 5.2.3.2.1, the lateness of the jobs is determined by using revised slack. By developing limitations (e.g., alternative 1, 2 and 3) on revised slack, it may not improve the solution, but it does allow the heuristic to search differently in many locals and may provide a better solution.

6.4.3 Slack Revision

The two methods used for revising slack (section 5.2.2) are presented and here tested their effectiveness. These two methods compute the slack as follows.

Method 1: $\mathrm{Slack}^{/}_{i,m} = d_i - \sum_{j \in m^{+}} p_{i,j} - \sum_{j \in m^{++}} q_{i,j}$

Method 2: $\mathrm{Slack}^{/}_{i,m} = d_i - \sum_{j \varepsilon m^{+}} p_{i,j} - s_{i,m^{+}} - \sum_{j \varepsilon m^{++}} q_{i,j}$

The percentage of problems, which the high priority job $L_{\max}$ of method 2 decreasing, increasing or tie from method 1, is used to compare performance.

In the no job priority case, the percentage of problems that both methods provides a different solution is relatively low compared to the two-job priority case, especially when the number of jobs is high as shown in figure 6-25 and 6-26. In both cases, the percentage that method 2 provides the higher or lower $L_{\max}$ is just slightly different in all scenarios. This indicates that neither method provides significantly better performance. This just shows another local search in which two methods can provide the different solutions.

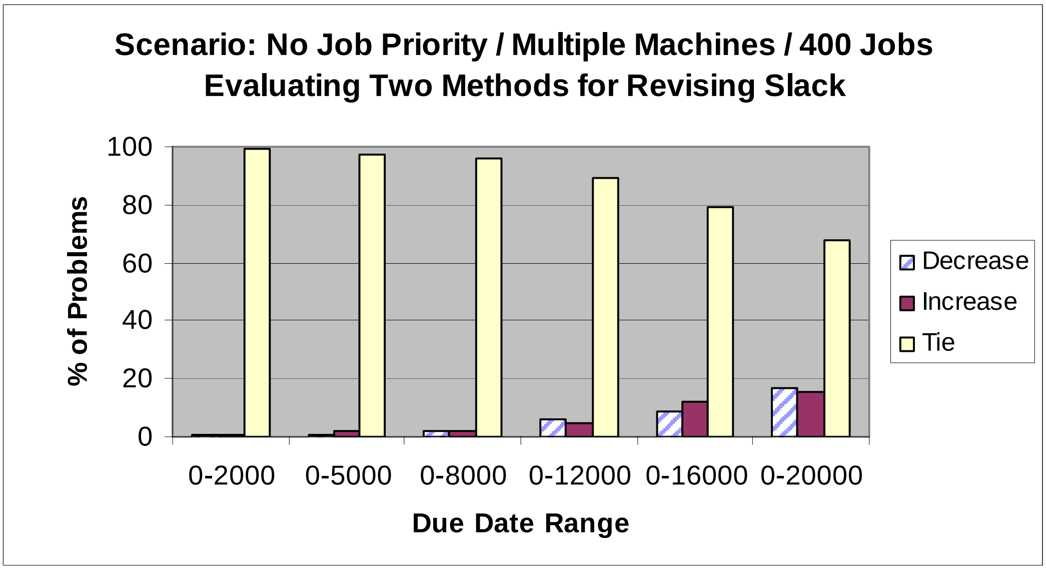


Figure 6-25: Evaluating Two Slack Revising Methods in No Job Priority Case

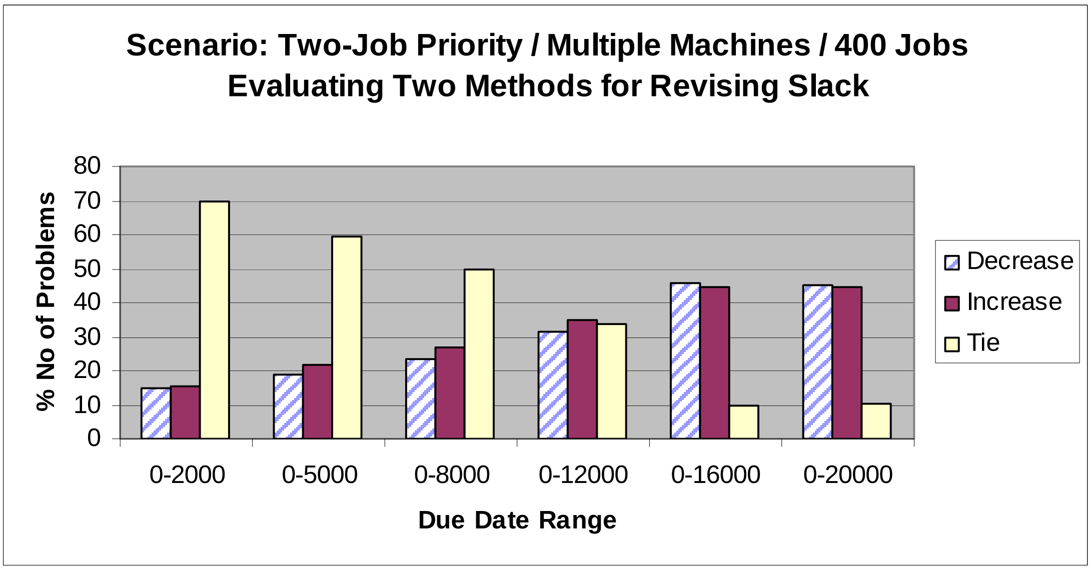


Figure 6-26: Evaluating Two Slack Revising Methods in Two-Job Priority Case

6.4.4 Varying the Number of Iterations

The algorithm was also tested for improvement by increasing the number of iterations (i.e., results from the algorithm run 500 iterations compared with results from the algorithm run 200 iterations).

As shown in table 6-8, increasing the number of iteration to 500 may improve the solution. However, there are scenarios where it doesn’t significantly improve the solution. For instance, the no-job priority, multiple machine operation with 400 jobs, or two-job priority, one machine operation with 100 jobs. The cells highlighted show scenarios where the percentage of problems improved by increasing the iterations are less than or equal to 10%.

Increasing the number of iterations seems to be more effective tool in the two-job priority classification case, specially when the production capacity is low (e.g., the one machine case).

Table 6-8: Percentage of Problems whose $L_{\max}$ ($HL_{\max}$) is Decreased by Increasing the Number of Iterations in Each Scenario

		No of Jobs	**Due Date Range**					
			0-2000	**0-5000**	**0-8000**	**0-12000**	**0-16000**	**0-20000**
No Job Priority Case	**One Machine Case**	**100**	8	14	24.5	44	57.5	42
		200	14.5	13	16.5	19	24.5	29.5
		400	12.5	10.5	9.5	10.5	8.5	16
	Multiple Machine Case	**100**	24	29.5	26	12.5	5.5	1
		200	14.5	18	17.5	24	40.5	47
		400	1	1	3	4	5.5	7.5
Two - Job Priority Case	**One Machine Case**	**100**	40	40	45.5	40	34.5	29
		200	36	42.5	43.5	54	55.5	57
		400	29	36	44.5	38	42.5	32.5
	Multiple Machine Case	**100**	15	10	5.5	3.5	0.5	0
		200	29.5	39	44.5	47	34	37.5
		400	20	21	26	37.5	45	51.5

6.4.5 Comparing with the Virtual Factory

The original virtual factory is modified and used for this comparison. Using the same scheduling problem but assuming no setup, the schedule and the $L_{\max}$ job can be found by applying the original Virtual Factory. Then by considering the schedule of the $L_{\max}$ job, the setups are inserted between two consecutive jobs, where the setup is required before processing the subsequent job. The new lateness of the $L_{\max}$ job from the Virtual Factory is re-calculated with the modified schedule and used for comparison.

Table 6-9: Percentage of Problems the Algorithm Developed Provides a Lower $L_{\max}$ ($HL_{\max}$) than the Virtual Factory

		No of Jobs	Due Date Range					
			0-2000	**0-5000**	**0-8000**	**0-12000**	**0-16000**	**0-20000**
No Job Priority Case	**One Machine Case**	**100**	93	91.5	84.5	81.5	92.5	91
		200	96.5	95	94.5	92.5	89.5	87.5
		400	98.5	97.5	99	97.5	98.5	98
	Multiple Machine Case	**100**	69.5	72	68	71.5	65	68.5
		200	69.5	74.5	68.5	59.5	73	72.5
		400	95	97	95	90	73	54
Two - Job Priority Case	**One Machine Case**	**100**	94	88.5	93.5	93.5	91.5	89.5
		200	93	95	92.5	84.5	85.5	90
		400	97.5	98	97	95.5	94	94
	Multiple Machine Case	**100**	79.5	68.5	69.5	70	60.5	62
		200	73	73.5	80	86	85	87.5
		400	72	70.5	75	75.5	79	78

Table 6-9 shows the percentage of problems that the new algorithm provides a lower $L_{\max}$ (or $HL_{\max}$) than the modified $L_{\max}$ from the Virtual Factory. The percentage is very high (e.g., 80% and above) in the one machine case. This may result from the fact that the number of jobs processed in each machine in the one machine case is greater than the multiple machine case, and thus require more setups.

6.4.6 Computational Times

Tables 6-10 and 6-11 show the average and maximum computational times for a method in solving a problem. In overall, the computational times of the two-job priority case are lower than the no-job priority case. This result is from the priority zone separation in the schedule, which reduces the number of positions tested for sequencing a family.

However, the computation time is quite small for all problem types. This provides the opportunity to apply many versions of the algorithm to get the best possible solution.

Table 6-10: Average Computational Time per Method per Problem (sec)

		No of Jobs	Due Date Range					
			0–2000	**0-5000**	**0-8000**	**0-12000**	**0-16000**	**0-20000**
No Job Priority Case	**One Machine Case**	**100**	0.60	0.77	0.84	0.88	0.84	0.80
		200	1.95	2.35	2.46	2.78	2.76	1.71
		400	4.10	4.69	4.94	5.54	5.52	11.14
	Multiple Machine Case	**100**	0.91	0.42	0.60	0.57	0.56	0.56
		200	0.95	1.77	1.05	1.87	1.73	1.74
		400	3.84	3.64	3.65	3.90	6.42	6.44
Two - Job Priority Case	**One Machine Case**	**100**	0.70	0.75	0.75	0.72	0.70	0.67
		200	2.19	1.48	2.21	2.20	2.21	2.18
		400	3.77	4.31	4.45	4.47	7.21	7.23
	Multiple Machine Case	**100**	0.64	0.81	0.65	0.62	0.55	0.55
		200	0.90	2.29	1.06	1.52	1.55	1.51
		400	2.58	2.87	2.98	3.06	5.07	4.26

Table 6-11: Maximum Computational Time per Method per Problem (sec)

		No of Jobs	Due Date Range					
			0-2000	**0-5000**	**0-8000**	**0-12000**	**0-16000**	**0-20000**
No Job Priority Case	**One Machine Case**	**100**	3.04	1.63	1.56	1.34	1.22	1.24
		200	6.70	6.31	5.81	5.28	5.55	3.20
		400	20.92	21.37	18.30	15.72	14.78	22.82
	Multiple Machine Case	**100**	4.57	0.66	0.88	0.83	0.87	0.82
		200	2.34	3.59	2.30	3.47	3.26	2.94
		400	9.05	8.19	7.86	8.06	12.15	13.25
Two - Job Priority Case	**One Machine Case**	**100**	1.33	1.20	1.14	1.19	1.19	1.15
		200	7.16	2.56	3.73	3.59	3.14	3.92
		400	9.63	9.34	9.23	8.73	13.37	13.31
	Multiple Machine Case	**100**	1.13	1.48	1.00	0.99	1.01	1.93
		200	1.62	7.64	1.59	2.38	2.27	2.39
		400	5.48	4.99	5.32	5.07	18.09	15.77

Chapter 7
Conclusions and Future Research

7.1 Conclusions

The dyeing and finishing scheduling problem, which is a job shop (flexible job shop) scheduling problem with sequence dependent setups, is studied. The fundamental structure of the scheduling algorithm developed is the Virtual Factory. Modifications are made in the Virtual Factory to schedule jobs where the operations can have either single or parallel machines, and some operations require sequence dependent setups. The scheduling problem is studied separately in two cases, no job priority and two-job priority classification. Scheduling heuristics are developed for solving the single and parallel machine problem with and without setups in both job priority cases.

In this dissertation, a fundamental scenario is presented from the textile industry. The model is based on the Virtual Factory. The scheduling problem is structured to resemble a real situation in industry. The setup and processing times, and the family groupings are based on real industry characteristics. All the information supplied here is based on the actual case plant.

A scheduling problem is solved by applying several versions of the scheduling heuristic and using the best solution found among them. The lower bound ("estimated lower bound") used, which does not take into account the setup, performs well if and only if the production situation is tight (e.g., due date ranges are small and production capacity is relatively low). It is clear that the lower bound used is ineffective in less tight production (e.g., due date ranges are high and production capacity is relatively high). Therefore, the lower bound used in this research is not necessarily an efficient tool for determining algorithm performance.

In the two-job classification case, the algorithm shows good performance. A high percentage of problems solved reach both the minimum high priority $L_{\max}$ and low priority $L_{\max}$.

The heuristic with the combination of 4 approaches (positioning, switching, $L_{\max}$ family splitting, and due after family splitting) seems to perform the best for all cases. In the large due date ranges, initially arranging jobs in due date related order obviously improves algorithm performance. There is no significant improvement for re-scheduling when a job leaves the queue.

Applying the different rules for moving the low priority $L_{\max}$ job and the different methods for revising the slack, there is no clear cut which one performs better than the others. However, each rule and each method lead the heuristic to the different local searches, which can provide different solutions. Thus a better solution can be found by applying more versions of the heuristic.

Although the first ten iterations of the algorithm normally obtain the best solution, it is not a particularly high percentage, especially in the large due date ranges. The experimental results show that increasing the number of iterations still can improve algorithm performance. Increasing the number of iteration is more effective for the two-priority classification problem.

The good performance of the algorithm is shown by its computation time. The algorithm is fast even for relatively large problems (i.e., industrial-sized). This is good, especially in dyeing and finishing processes, where unexpected events (i.e., rework, rush order, setup) occur all the time. The capability for re-scheduling is important.

The short computational times also provide an opportunity for improving algorithm performance. An important characteristic of the algorithm is that it can be modified easily. The experimentation indicates that solutions may be improved by developing simple modifications to existing versions of the algorithm. Computational efficiency allows the application of a number of versions within a reasonable amount of computation time. The observations provided in this dissertation are useful for determining the right versions for the right problem.

7.2 Future Research

A high priority should be placed on strengthening the lower bound. Although it is the true lower bound for the single machine problem, it is not a tight lower bound because the setup is not taken into account. For the two-priority case, the current bound used is only an estimate for both single machine and parallel machine problem, not a true lower bound. Obtaining a good theoretical lower bound seems extremely difficult but it may be possible to construct. Improvement of the lower bound can improve the performance of the algorithm.

Although the scheduling algorithms are tested on a wide range of artificially generated scenarios, it is important to obtain industrial data to further validate performance. For doing this, the simplified structures used for computing setup and processing times, and family groupings need to be verified using real data.

Other scheduling heuristics can be developed and coupled with the Virtual Factory. The modified version in the original Virtual Factory, inserting idle time before the hot job begins processing [19], can also be implemented and its effectiveness tested in this type of problem.

All the sequence dependent setup operations right now are treated as a single machine problem. The parallel machine problem with sequence dependence, where a job can be processed in any machine, is another interesting area. Solving this problem can reduce the limitation of this technological constraint.

Other scheduling objectives can be studied in this job shop problem with setup. Minimizing the setup times and the idle times (in some machines), the mean flow times or the maximum completion times are also the interesting objectives for the dyeing and finishing process. Fortunately, with a proper scheduling heuristic, all of these objectives can be solved with the Virtual Factory.

The next level in implementing the Virtual Factory, where the job allocation is included as a function for solving a scheduling problem, is very challenging. Changing the job production routes, which means the change in job allocation, is sensitive to the solution in the job shop problem. The advantage of the Virtual Factory, which is a simulation based approach, provides an opportunity to combine these two functions together. The idea is to combine job allocation and job sequencing to achieving a scheduling objective. Workload balancing for minimizing the maximum completion time (or idle time), or family allocation for minimizing the setup time in job allocation may also aid job sequencing to achieve a scheduling objective (i.e., minimizing $L_{\max}$).

Bibliography

[1]. Adams J., Balas E, and Zawack D., (1988). “The Shifting Bottleneck Procedure for Job Shop Scheduling,” Management Science, Vol. 34, 391-401.

[2]. Allahverdi A.; Gupta J. N. D; and Aldowaisan T., (1999). “A Review of Scheduling Research Involving Setup Consideration,” Omaga, The internation journal of management science, 27, 219-239.

[3]. Baker, K. R. and Su, Z. (1974). “Sequencing with Due-Dates and Early Start Times to Minimize Maximum Tardiness,” Naval Research Logistics Quarterly, 21:1, 171-176.

[4]. Balas E., Lancia G., Serafini P., and Vazacopoulos A., (1998). “Job Shop Scheduling with Deadlines,” Journal of Combinatorial Optimization, Vol. 1, 329-353.

[5]. Bianco L., Ricciardelli S., Rinaldi G., and Sassano A., (1988). “Scheduling Tasks with Sequence Dependent Processing times,” Naval Research Logistics, Vol. 35, 177-184.

[6]. Bitran G. R. and Gilbert S. M., (1990). “Sequencing Production on Parallel Machines with Two Magnitudes of Sequence Dependent Setup Cost,” Journal of Manufacturing and Operations Management, Vol. 3, 24-52.

[7]. Blazewicz, J. (1977). “Simple Algorithms for Multiprocessor Scheduling to Meet Deadlines,” Information Processing Letters, 6:5, 162-164.

[8]. Bruno J., and Downey P., (1978). “Complexity of Tasks Sequencing with Deadlines, Setup Times, and Changeover Costs,” SIAM Journal of Computing, Vol. 7, 393-404.

[9]. Carlier, J. and Pinson, E., (1989). “An Algorithm for Solving the Job Shop Problem,” Management Science, 35, 164-176.

[10]. Choi I. C. and Korkmaz O., (1997). “Job Shop Scheduling with Separable Sequence Dependent Setups,” Annuals of Operations Research, Vol. 70, 155-170.

[11]. Corwin B. D. and Esogbue A. O., (1974). “Two Machine Flow Shop Scheduling Problems with Sequence Dependent Setup Times: a Dynamic Programming Approach,” Nav Res Logistics Q, Vol. 21, No. 3, Sep, 515-524.

[12]. Das S. R., Gupta J. N. D., and Khumawala B. M., (1995). “A Savings Index Heuristic Algorithm for Flowshop Scheduling with Sequence Dependent Setup Time,” The Journal of the Operational Research Society, Vol. 46, Issue 11, Nov., 1365-1373.

[13]. Deane R. H. and White E. R., (1975). “Balancing Workloads and Minimizing Setup Costs in the Parallel Processing Shop,” Operational Research Quarterly, Vol. 26, 45-53.

[14]. Flynn B. B., (1978). “The Effects of Setup Time on Output Capacity in Cellular Manufacturing,” International Journal of Production Research, Vol. 25, 1761-1772.

[15]. Franca P. M., Gendreau M., Laporte G., and Muller F. M., (1996).” A Tabu Search Heuristic for the Multiprocessor Scheduling Problem with Sequence Dependent Setup Time,” International Journal of Production Economics, 43, 79-89.

[16]. Gupta S. K., (1982). “N Jobs and M Machines Job-Shop Problems with Sequence Dependent Setup Times,” Int. J Prod. Res., Vol. 2, No. 5, 643-656.

[17]. Gupta S. K. and Darrow W. P., (1986). “The Two-Machine Sequence Dependent Flowshop Scheduling Problem,” European Journal of Operational Research, Vol. 24, 439-446.

[18]. Hodgson T. J., Cormier D., Weintraub A. J., and Zozom Jr A., (1998). “Note. Satisfying Due Dates in Large Job Shop,” Management Science, Vol. 44, No. 10, Oct, 1442-1446.

[19]. Hodgson T. J., King R. E., Thoney K. A., Stanislaw N., Weintraub A., and Zozom A., (1999). “On Satisfyign Due-Dates in Large Job Shops: Idle Time Insertion,” IIE Transactions 32:2, 177-180.

[20]. Horn, W.A. (1974). “Some Simple Scheduling Algorithms,” Naval Research Logistics Quarterly, 21:1, 177-185.

[21]. Hwang H. and Sun J. U., (1998). “Production Sequencing Problem with Re-entrant Workflows and Sequence Dependent Setup Times,” Int. J. Prod. Res., Vol. 36, No. 9, 2435-2450.

[22]. Jensen J. B., Malhotra M. K., and Philipoom P. R., (1998). “Family-Based Scheduling of Shops with Functional Layouts,” International Journal of Production Research, Vol. 36, 177-180.

[23]. Kang S., Malik K., and Thomas L. J., (1999). “Lotsizing and Scheduling on Parallel Machines with Sequence Dependent Setup Costs,” Management Science, Vol. 45, 273-289.

[24]. Kim S. C. and Bobrowski P. M., (1994). “Impact of Sequence-Dependent Setup Time on Job shop Scheduling Performance,” Int. J. Prod. Res., Vol. 32, No. 7, 1503-1520.

[25]. Krajewski L. J., King B. E., Ritzman L. P., and Wong D. S., (1987) “Kanban, MRP, and Shaping the Manufacturing environment,” Management Science, Vol. 33, 39-57.

[26]. Labetoulle, J., Lawler, E. L., Lenstra, J. K., and Kan, A. H. G. (1984). “Preemptive Scheduling of Uniform Machines Subject to Release Dates,” Progress in Combinatorial Optimization, Academic Press, 245-261.

[27]. Lawrence S. R. and Morton T. E., (1986). “Patriarch: Hierarchical Production Scheduling,” National Bureau pf Standards Special Publication, Vol. 724, 87-97.

[28]. Livingston D. L. and Sommerfeld J. T., (1989). “Discrete-Event Simulation in the Design of Textile Finishing Processes,” Textile Research Institute, Oct., 589-596.

[29]. Mahmoodi F., Dooley K. J., Starr P. J., (1990). “An Investigation of Dynamic Group Scheduling Heuristics in a Job Shop Manufacturing Cell,” International Journal of Production Research, Vol. 28, 1695-1711.

[30]. Maldonado F., Ciurlizza A., Radillo R., and E Ponce de León, (2000). “Optimization of the Color Sequence in the Dyeing Process: Industrial Application,” JSDC, Vol. 116, Nov., 359-362.

[31]. Marsh J. D. and Montgomery D. C., (1973). “Optimal Procedures for Scheduling Jobs with Sequence Dependent Changeover Times on Parallel Processors,” AIIE technical Papers, 279-286.

[32]. Monma C. L., and Potts C. N., (1989). “On the Complexity of Scheduling with Batch Setup Times,” Operations Research, Vol. 37, Issue 5, 798-804.

[33]. Morales L., Maldonado F., Radillo R., and Ciurlizza A., (1996). “Optimization of the Color Sequence in the Process of Fabric Dyeing,” JSDC, Vol. 112, Dec., 361-363.

[34]. Ovacik I. M. and Uzsoy R., (1993). “Worst-Case Error Bounds for Parallel Machine Problems with Bounded Sequence Dependent Setup Times,” Operations Research Letters, Vol. 14, 251-256.

[35]. Ovacik I. M. and Uzsoy R., (1994). “Exploiting Shop Floor Status Information to Schedule Complex Job Shops,” Journal of Manufacturing Systems, Vol. 13, 73-84.

[36]. Panwalker S. S., Dudek R. A., and Smith M. L., (1973). “Sequencing Research and the Industrial Scheduling Problem,” (Elmahgrahby S. E. , editor), Symposium on the Theory of Scheduling and its Applications, 29-38.

[37]. Parthasarathy S. and Rajendran C., (1997). “A Simulated Annealing Heuristic for Scheduling to Minimize Weighted Tardiness in a Flowshop with Sequence Dependent Setup Times of Jobs – a Case Study,” Production Planning and Control, Vol. 8, 475-483.

[38]. Parthasarathy S. and Rajendran C., (1997). “An Experimental Evaluation of Heuristics for Scheduling in a Real-Life Flowshop with Sequence Dependent Setup Times of Jobs,” International Journal of Production Economics, Vol. 49, 255-263.

[39]. Pearn W. L., Chung S. H., and Yang M. H., (2002). “A Case Study on the Wafer Probing Scheduling Problem,” Production Planning & Control, Vol. 13, No. 1, 66-75.

[40]. Rajendran C. and Ziegler H., (1997). "A Heuristic for Scheduling to Minimize the Sum of Weighted Flowtime of Jobs in a Flowshop with Sequence Dependent Setup Times of Jobs," Computers and Industrial Engineering, Vol. 33, 281-284.

[41]. Rios-Mercado R. Z. and Bard J. F., (1999). "An enhanced TSP-Based Heuristic for Makespan Minimization in a Flowshop with Setup times," Journal of Heuristics, Vol. 5, 53-70.

[42]. Sahni, S. (1979). "Preemptive Scheduling with Due Dates," Operations Research, 27, 925-934.

[43]. Schultz, S. R., Hodgson T. J., King R. E., and Taner M. R. "Technical Note: Minimizing Lmax for the Single Machine Scheduling Problem with Family Setups," Submitted to Int J Prod Res

[44]. Srikar B. N. and Ghosh S., (1986). "A MILP Model for n-Job, M-Stage Flow Shop," International Journal of Production Research, Vol 24, No 6, 1459-1472.

[45]. Stafford E. F. and Tseng F. T., (1990). "On the Srikar-Ghosh MILP Model for the $N \times M$ SDST Flowshop Problem," International Journal of Production Research, Vol. 28, 1817-1830.

[46]. Sun X. Q. and Noble J. S., (1999). "An Approach to Job Shop Scheduling with Sequence Dependent Setups," Journal of Manufacturing Systems, Vol. 18, 416-430.

[47]. Szwarc W. and Gupta J. W. D., (1987). " A Flowshop Problem with Sequence Dependent Setup Times," Naval Research Logistics Quarterly, Vol. 34, 619-634.

[48]. Taner M. R., "Scheduling with Sequence Dependent Setup Times," Unpublished Ph.D. dissertation, North Carolina State University.

[49]. Thoney, K. A., "Simultaneous Plant and supply Chain SCheduling," Unpublished Ph.D. dissertation, North Carolina State University.

[50]. Tucci M. and Rinaldi R., (1999). "From Theory to Application: Tabu Search in Textile Production Scheduling," Production Planning & Control, Vol. 10, No. 4, 365-374.

[51]. Uskup G. and Smith S. H., (1975). "A Branch and Bound for Two-Stage Production Sequencing," Operation Research, Vol. 23, 118-136.

[52]. Vepsalainen A. and Morton T. E., (1987). "Priority Rules and Leadtime Estimation for Job Shop Scheduling with Weighted Tardiness Costs," Management Science, Vol. 33, 1036-1047.

[53]. Weintraub A., Cormier D., Hodgson T. J., King R. E., Wilson J. R., and Zozom A., (1999). "Scheduling with Alternatives: A Link between Process Planning and Scheduling," IIE Transactions, 31:11, 1093-1102.

[54]. Werner F. and Winkler A., (1995). “Insertion Techniques for the Heuristic Solution of the Job Shop Problem,” Discrete Applied Mathematics, Vol. 58, 191-211.

[55]. White C. H., and Wilson R. C., (1977). “Sequence Dependent Set-up Times and Job Sequencing,” Int. J. Prod. Res., Vol.15, No. 2, 191-202.

[56]. Zhou C. and Egbelu P. J., (1989). “Scheduling in a Manufacturing Shop with Sequence Dependent Setups,” Robotics and Computer Integrated Manufacturing, Vol. 5, 73-81.

Appendices

Appendix A: Statistical Multiple Means Comparison

The means of $L_{\max}$ of the different scheduling methods considered are compared using four statistical comparison tests, which are based on the analysis of variance (ANOVA) at $\alpha > 0.01$ (99%).

The four methods of comparison, each performing a multiple means comparison, are Tukey, which is based on the studentized range distribution (standardized maximum difference between the means), Student-Newman-Keuls (SNK), which is a multiple range test using a multiple stage approach, the Duncan, which tries to minimize the Bayesian loss function, and an approach developed by Ryan, Einot and Gabriel, and Welsch (REGW) which is also a multiple stage approach which controls the maximum experiment-wise error rate under any complete or partial hypothesis.

Nine different scheduling methods are tested in the no job priority case, and twelve different scheduling methods are tested in the two-job priority case. The multiple means tests provide information on sets of means of $L_{\max}$ ($HL_{\max}$) whose differences are statistically significant.

As shown in the result tables, in each test the scheduling methods are grouped based on the difference of their means. The alphabet (i.e., A, B, C, D) is used to indicate the methods which are in the same group. If the methods have the same letter, they are in the same group, which means there is no statistically significant difference in the means. If the methods have different letters, they are in a different group, which means there is a statistically significant difference in the means. However, two groups can be overlapping (i.e., two groups have some methods in common). In the tables, the rank indicates the ranking of the average $L_{\max}$

($HL_{\max}$), where the ranking is from lowest (left) to (highest) right. The test results are shown in the following tables.

- Table A-1 – A-3 are the results from no job priority with one machine
- Table A-4 – A-6 are the results from no job priority with multiple machines
- Table A-7 – A-9 are the results from two-job priority with one machine
- Table A-10 – A-12 are the results from two job priority with multiple machines

Table A-1: Multiple Means Comparison for Scheduling Methods in
Scenario: No Job Priority / One Machine / 100 Jobs

No of	Due Date	Multiple Means Comparison Between 9 Scheduling Methods (M1 - M9)									
Jobs	Ranges	Scenario : No Job Priority Classification / One Machine Case									
100	0 - 2000	**Rank**	**M6**	**M8**	**M3**	**M7**	**M9**	**M4**	**M5**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M7**	**M4**	**M6**	**M8**	**M9**	**M5**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M7**	**M4**	**M8**	**M9**	**M6**	**M3**	**M5**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M7**	**M4**	**M8**	**M9**	**M6**	**M5**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	
					B	B	B	B	B	B	B
		SNK	A	A	A	A	A	A	A	A	
					B	B	B	B	B	B	B
		Duncan	A	A	A	A	A	A	A	A	
										B	B
		REGW	A	A	A	A	A	A	A	A	
					B	B	B	B	B	B	B
	0 - 16000	**Rank**	**M7**	**M4**	**M8**	**M9**	**M6**	**M5**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	
							B	B	B	B	B
		SNK	A	A	A	A	A	A	A	A	
										B	B
		Duncan	A	A	A	A	A	A	A	A	
										B	B
		REGW	A	A	A	A	A	A	A	A	
										B	B
	0 - 20000	**Rank**	**M4**	**M7**	**M8**	**M9**	**M6**	**M5**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	
										B	B
		SNK	A	A	A	A	A	A	A	A	
										B	B
		Duncan	A	A	A	A	A	A	A	A	
										B	B
		REGW	A	A	A	A	A	A	A	A	
										B	B

Table A-2: Multiple Means Comparison for Scheduling Methods in
Scenario: No Job Priority / One Machine / 200 Jobs

No of Jobs	**Due Date Ranges**	**Multiple Means Comparison Between 9 Scheduling Methods (M1 - M9) Scenario : No Job Priority Classification / One Machine Case**									
200	**0 - 2000**	**Rank**	**M6**	**M9**	**M8**	**M1**	**M3**	**M7**	**M2**	**M4**	**M5**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M6**	**M8**	**M9**	**M3**	**M5**	**M7**	**M1**	**M4**	**M2**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M8**	**M9**	**M6**	**M7**	**M3**	**M5**	**M4**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M7**	**M4**	**M9**	**M6**	**M8**	**M5**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 16000	**Rank**	**M7**	**M4**	**M6**	**M8**	**M9**	**M5**	**M2**	**M3**	**M1**
		Tukey	A	A	A	A	A	A	A	A	
				B	B	B	B	B	B	B	B
		SNK	A	A	A	A	A	A	A	A	
				B	B	B	B	B	B	B	B
		Duncan	A	A	A	A	A	A	A	A	
				B	B	B	B	B	B	B	B
		REGW	A	A	A	A	A	A	A	A	
				B	B	B	B	B	B	B	B
	0 - 20000	**Rank**	**M7**	**M4**	**M9**	**M8**	**M6**	**M2**	**M5**	**M3**	**M1**
		Tukey	A	A	A	A	A	A	A	A	
										B	B
		SNK	A	A	A	A	A	A	A	A	
											B
		Duncan	A	A	A	A	A				
				B	B	B	B	B	B	B	
											C
		REGW	A	A	A	A	A	A	A	A	
											B

Table A-3: Multiple Means Comparison for Scheduling Methods in
Scenario: No Job Priority / One Machine / 400 Jobs

No of Jobs	Due Date Ranges	Multiple Means Comparison Between 9 Scheduling Methods (M1 - M9) Scenario : No Job Priority Classification / One Machine Case									
400	0 - 2000	**Rank**	**M1**	**M2**	**M6**	**M3**	**M7**	**M9**	**M8**	**M4**	**M5**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M1**	**M2**	**M6**	**M9**	**M7**	**M8**	**M3**	**M5**	**M4**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M6**	**M9**	**M8**	**M1**	**M5**	**M3**	**M7**	**M2**	**M4**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M8**	**M6**	**M9**	**M7**	**M4**	**M3**	**M2**	**M5**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 16000	**Rank**	**M6**	**M7**	**M8**	**M9**	**M4**	**M3**	**M5**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 20000	**Rank**	**M6**	**M8**	**M3**	**M7**	**M9**	**M4**	**M5**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A

Table A-4: Multiple Means Comparison for Scheduling Methods in Scenario: No Job Priority / Multiple Machines / 100 Jobs

No of Jobs	Due Date Ranges	Multiple Means Comparison Between 9 Scheduling Methods (M1 - M9) Scenario : No Job Priority Classification / Multiple Machine Case									
100	0 - 2000	**Rank**	**M6**	**M9**	**M8**	**M7**	**M3**	**M5**	**M4**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M9**	**M8**	**M7**	**M6**	**M5**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M7**	**M4**	**M8**	**M6**	**M9**	**M3**	**M5**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M7**	**M4**	**M6**	**M8**	**M3**	**M9**	**M5**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 16000	**Rank**	**M7**	**M4**	**M5**	**M8**	**M3**	**M6**	**M2**	**M9**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 20000	**Rank**	**M6**	**M7**	**M3**	**M4**	**M5**	**M8**	**M9**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A

Table A-5: Multiple Means Comparison for Scheduling Methods in
Scenario: No Job Priority / Multiple Machines / 200 Jobs

No of Jobs	**Due Date Ranges**	**Multiple Means Comparison Between 9 Scheduling Methods (M1 - M9) Scenario : No Job Priority Classification / Multiple Machine Case**									
200	**0 - 2000**	**Rank**	**M9**	**M7**	**M8**	**M5**	**M6**	**M3**	**M4**	**M1**	**M2**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M8**	**M9**	**M6**	**M5**	**M7**	**M3**	**M4**	**M1**	**M2**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M7**	**M9**	**M8**	**M6**	**M4**	**M5**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M7**	**M9**	**M8**	**M6**	**M4**	**M5**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	
								B	B	B	B
		SNK	A	A	A	A	A	A	A	A	
									B	B	B
		Duncan	A	A	A	A	A	A	A	A	
										B	B
		REGW	A	A	A	A	A	A	A	A	
								B	B	B	B
	0 - 16000	**Rank**	**M7**	**M4**	**M8**	**M9**	**M6**	**M5**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	
										C	C
		SNK	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	
											C
		Duncan	A	A	A	A	A	A	A		
							B	B	B	B	
											C
		REGW	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	
										C	C
	0 - 20000	**Rank**	**M7**	**M9**	**M4**	**M8**	**M6**	**M5**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	
										C	C
		SNK	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	
										C	C
		Duncan	A	A	A	A	A	A	A		
								B	B	B	
										C	C
		REGW	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	
										C	C

Table A-6: Multiple Means Comparison for Scheduling Methods in Scenario: No Job Priority / Multiple Machines / 400 Jobs

No of Jobs	Due Date Ranges	Multiple Means Comparison Between 9 Scheduling Methods (M1 - M9) Scenario : No Job Priority Classification / Multiple Machine Case									
400	0 - 2000	Rank	M4	M5	M3	M2	M1	M7	M8	M6	M9
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 5000	Rank	M5	M3	M8	M1	M9	M4	M6	M7	M2
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 8000	Rank	M5	M8	M4	M6	M7	M3	M9	M2	M1
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 12000	Rank	M7	M8	M9	M4	M6	M5	M3	M2	M1
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 16000	Rank	M7	M6	M8	M9	M4	M5	M3	M2	M1
		Tukey	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A
	0 - 20000	Rank	M7	M8	M9	M6	M4	M5	M3	M2	M1
		Tukey	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	
									C	C	C
		SNK	A	A	A	A	A	A			
				B	B	B	B	B	B	B	
										C	C
		Duncan	A	A	A	A	A	A			
				B	B	B	B	B	B		
							C	C	C	C	
										D	D
		REGW	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	
									C	C	C

Table A-7: Multiple Means Comparison for Scheduling Methods in Scenario: Two-Job Priority / One Machine / 100 Jobs

No of Jobs	Due Date Ranges	Multiple Means Comparison Between 12 Scheduling Methods (M1 - M12) Scenario : Two - Job Priority Classification / One Machine Case												
100	**0 - 2000**	**Rank**	**M7**	**M11**	**M10**	**M12**	**M4**	**M8**	**M5**	**M9**	**M6**	**M1**	**M2**	**M3**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M12**	**M10**	**M11**	**M8**	**M7**	**M9**	**M5**	**M6**	**M4**	**M2**	**M3**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M9**	**M6**	**M10**	**M5**	**M12**	**M11**	**M4**	**M8**	**M7**	**M2**	**M3**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M5**	**M10**	**M9**	**M8**	**M6**	**M4**	**M11**	**M7**	**M12**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 16000	**Rank**	**M5**	**M10**	**M6**	**M9**	**M4**	**M8**	**M11**	**M12**	**M7**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 20000	**Rank**	**M6**	**M9**	**M5**	**M8**	**M4**	**M10**	**M7**	**M11**	**M12**	**M2**	**M3**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A

Table A-8: Multiple Means Comparison for Scheduling Methods in
Scenario: Two-Job Priority / One Machine / 200 Jobs

No of Jobs	Due Date Ranges	Multiple Means Comparison Between 12 Scheduling Methods (M1 - M12) Scenario : Two - Job Priority Classification / One Machine Case											
200	0 - 2000	**Rank**	**M11**	**M12**	**M8**	**M4**	**M10**	**M1**	**M7**	**M9**	**M6**	**M5**	**M3**	**M2**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M12**	**M8**	**M10**	**M11**	**M9**	**M5**	**M6**	**M4**	**M7**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M10**	**M9**	**M5**	**M11**	**M6**	**M12**	**M8**	**M7**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M9**	**M10**	**M6**	**M5**	**M12**	**M11**	**M8**	**M7**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	
				B	B	B	B	B	B	B	B	B	B	B
		SNK	A	A	A	A	A	A	A	A	A	A	A	
				B	B	B	B	B	B	B	B	B	B	B
		Duncan	A	A	A	A	A	A	A	A	A	A	A	
											B	B	B	B
		REGW	A	A	A	A	A	A	A	A	A	A	A	
				B	B	B	B	B	B	B	B	B	B	B
	0 - 16000	**Rank**	**M9**	**M10**	**M6**	**M5**	**M12**	**M11**	**M8**	**M7**	**M4**	**M2**	**M3**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	
											B	B	B	B
		SNK	A	A	A	A	A	A	A	A	A	A	A	
												B	B	B
		Duncan	A	A	A	A	A	A	A	A	A	A	A	
												B	B	B
		REGW	A	A	A	A	A	A	A	A	A	A	A	
												B	B	B
	0 - 20000	**Rank**	**M6**	**M10**	**M5**	**M9**	**M12**	**M11**	**M8**	**M7**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	
				B	B	B	B	B	B	B	B	B	B	B
		SNK	A	A	A	A	A	A	A	A	A	A	A	
							B	B	B	B	B	B	B	B
		Duncan	A	A	A	A	A	A	A	A	A	A	A	
											B	B	B	B
		REGW	A	A	A	A	A	A	A	A	A	A	A	
					B	B	B	B	B	B	B	B	B	B

Table A-9: Multiple Means Comparison for Scheduling Methods in Scenario: Two-Job Priority / One Machine / 400 Jobs

No of Jobs	Due Date Ranges	Multiple Means Comparison Between 12 Scheduling Methods (M1 - M12) Scenario : Two - Job Priority Classification / One Machine Case												
400	0 - 2000	**Rank**	**M4**	**M8**	**M9**	**M12**	**M7**	**M1**	**M3**	**M11**	**M2**	**M6**	**M10**	**M5**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M11**	**M12**	**M8**	**M9**	**M5**	**M10**	**M6**	**M7**	**M4**	**M2**	**M1**	**M3**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M8**	**M10**	**M9**	**M11**	**M12**	**M6**	**M5**	**M7**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M10**	**M9**	**M6**	**M5**	**M2**	**M3**	**M12**	**M8**	**M11**	**M4**	**M7**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 16000	**Rank**	**M10**	**M9**	**M6**	**M5**	**M11**	**M12**	**M3**	**M8**	**M2**	**M7**	**M4**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 20000	**Rank**	**M10**	**M9**	**M6**	**M5**	**M12**	**M11**	**M3**	**M8**	**M2**	**M7**	**M4**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	
							B	B	B	B	B	B	B	B
		SNK	A	A	A	A	A	A	A	A	A	A	A	
							B	B	B	B	B	B	B	B
		Duncan	A	A	A	A	A	A	A	A	A	A	A	
														B
		REGW	A	A	A	A	A	A	A	A	A	A	A	
							B	B	B	B	B	B	B	B

Table A-10: Multiple Means Comparison for Scheduling Methods in
Scenario: Two-Job Priority / Multiple Machines / 100 Jobs

No of Jobs	**Due Date Ranges**	**Multiple Means Comparison Between 12 Scheduling Methods (M1 - M12) Scenario : Two - Job Priority Classification / Multiple Machine Case**												
100	**0 - 2000**	**Rank**	**M9**	**M8**	**M6**	**M4**	**M5**	**M7**	**M10**	**M11**	**M12**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M5**	**M9**	**M10**	**M6**	**M8**	**M4**	**M11**	**M12**	**M7**	**M2**	**M3**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M5**	**M8**	**M9**	**M4**	**M6**	**M10**	**M11**	**M12**	**M7**	**M1**	**M2**	**M3**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M9**	**M10**	**M5**	**M8**	**M6**	**M4**	**M12**	**M11**	**M7**	**M1**	**M3**	**M2**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 16000	**Rank**	**M8**	**M5**	**M10**	**M4**	**M9**	**M6**	**M11**	**M12**	**M7**	**M3**	**M1**	**M2**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 20000	**Rank**	**M4**	**M8**	**M9**	**M10**	**M6**	**M5**	**M2**	**M11**	**M12**	**M3**	**M7**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A

Table A-11: Multiple Means Comparison for Scheduling Methods in
Scenario: Two-Job Priority / Multiple Machines / 200 Jobs

No of Jobs	Due Date Ranges	Multiple Means Comparison Between 12 Scheduling Methods (M1 - M12) Scenario : Two - Job Priority Classification / Multiple Machine Case												
200	0 - 2000	**Rank**	**M12**	**M7**	**M11**	**M8**	**M4**	**M9**	**M10**	**M6**	**M5**	**M1**	**M3**	**M2**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M9**	**M12**	**M10**	**M11**	**M8**	**M6**	**M7**	**M5**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M10**	**M9**	**M6**	**M8**	**M12**	**M5**	**M11**	**M7**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M10**	**M9**	**M6**	**M5**	**M12**	**M8**	**M4**	**M7**	**M11**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 16000	**Rank**	**M8**	**M4**	**M9**	**M6**	**M5**	**M10**	**M11**	**M12**	**M7**	**M2**	**M3**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 20000	**Rank**	**M10**	**M9**	**M8**	**M4**	**M5**	**M6**	**M7**	**M11**	**M12**	**M1**	**M2**	**M3**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A

Table A-12: Multiple Means Comparison for Scheduling Methods in
Scenario: Two-Job Priority / Multiple Machines / 400 Jobs

No of Jobs	Due Date Ranges	Multiple Means Comparison Between 12 Scheduling Methods (M1 - M12) Scenario : Two - Job Priority Classification / Multiple Machine Case											
400	0 - 2000	**Rank**	**M7**	**M11**	**M12**	**M8**	**M4**	**M10**	**M9**	**M6**	**M5**	**M1**	**M3**	**M2**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 5000	**Rank**	**M11**	**M12**	**M7**	**M8**	**M10**	**M4**	**M9**	**M6**	**M5**	**M1**	**M3**	**M2**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 8000	**Rank**	**M9**	**M10**	**M11**	**M12**	**M8**	**M7**	**M6**	**M5**	**M4**	**M2**	**M3**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	A
		SNK	A	A	A	A	A	A	A	A	A	A	A	A
		Duncan	A	A	A	A	A	A	A	A	A	A	A	A
		REGW	A	A	A	A	A	A	A	A	A	A	A	A
	0 - 12000	**Rank**	**M10**	**M9**	**M12**	**M6**	**M11**	**M5**	**M8**	**M7**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	
					B	B	B	B	B	B	B	B	B	B
		SNK	A	A	A	A	A	A	A	A	A	A	A	
					B	B	B	B	B	B	B	B	B	B
		Duncan	A	A	A	A	A	A	A	A	A	A	A	
											B	B	B	B
		REGW	A	A	A	A	A	A	A	A	A	A	A	
					B	B	B	B	B	B	B	B	B	B
	0 - 16000	**Rank**	**M10**	**M9**	**M12**	**M11**	**M6**	**M5**	**M8**	**M7**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	
												B	B	B
		SNK	A	A	A	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	B	B	B	
												C	C	C
		Duncan	A	A	A	A	A	A	A	A	A			
					B	B	B	B	B	B	B	B		
								C	C	C	C	C	C	
												D	D	D
		REGW	A	A	A	A	A	A	A	A	A	A		
				B	B	B	B	B	B	B	B	B	B	
												C	C	C
	0 - 20000	**Rank**	**M10**	**M9**	**M5**	**M6**	**M11**	**M12**	**M8**	**M7**	**M4**	**M3**	**M2**	**M1**
		Tukey	A	A	A	A	A	A	A	A	A	A	A	
												B	B	B
		SNK	A	A	A	A	A	A	A	A	A	A	A	
												B	B	B
		Duncan	A	A	A	A	A	A	A	A	A			
					B	B	B	B	B	B	B	B		
							C	C	C	C	C	C	C	
												D	D	D
		REGW	A	A	A	A	A	A	A	A	A	A	A	
												B	B	B

Appendix B: Abbreviations of Fabric Types

Poly: Polyester fabric

TK: Staple polyester fabric

TC: Polyester cotton blended fabric, which the percentage of polyester composition higher than cotton

CVC: Polyester cotton blended fabric, which the percentage of cotton composition higher than polyester

TR: Polyester rayon blended fabric