ABSTRACT

CHO, EUNKYOUNG GLORIA. Scheduling Supply Chains with Batchwise Fabric Dyeing Operations. (Under the direction of Kristin A. Thoney and Russell E. King)

Meeting customer due dates has become important for textile coloration firms' long-term survival due to rapidly changing business conditions and intense global competition. In this dissertation, optimization of the fiber-textile-apparel-retail chain, including batchwise fabric dyeing operations, was pursued.

The performance of the Virtual Factory (VF), a job shop scheduling system developed at North Carolina State University, was tested in multi-factory, rolling horizon settings to more accurately predict how it would perform in industry by eliminating transient effects presented in previous experimentation. The VF performed well in all multi-factory supply chain environments.

By taking the theoretical approach of color physics, setup matrices for dyeing operations were developed to include four indices; fabric/dye type, hue, lightness and chroma. After refining the matrices to capture the interdependency of the hue, lightness, and chroma of colors dyed within the same fabric/dye type, an existing sequence dependent scheduling algorithm was modified accordingly. The proposed algorithm and additional modifications were implemented in the VF, and a variety of one machine flowshop scenarios were tested against another algorithm found in the scheduling literature and a lower bound approximation. The proposed algorithm performed well in tight due date ranges and with a large number of jobs but not as well under other conditions. Sensitivity analysis of the initial parameters used in the proposed algorithm showed that its performance is highly dependent on these values.

SCHEDULING SUPPLY CHAINS
WITH BATCHWISE FABRIC DYEING OPERATIONS

by
EUNKYOUNG GLORIA CHO

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

TEXTILE TECHNOLOGY MANAGEMENT

Raleigh

2004

APPROVED BY:

Kristin A. Thoney (Co-chair)	Russell E. King (Co-chair)
Trevor J. Little	Thom J. Hodgson

BIOGRAPHY

Eunkyoung Cho (also known as Gloria), only child of Tae-Yong Cho and Young-Mi Choi, was born April 1st, 1974. She was raised in Seoul, South Korea. Eunkyoung graduated from Sook-Myoung Women's High School in 1993. In 1997, she graduated with honors from Seoul National University in Seoul, South Korea and received her B.S. in Fiber and Polymer Science. In 1999, Eunkyoung completed her M.S. degree in Fiber and Polymer Science at the same university and the theme of her thesis was color physics.

In August of 2000, Eunkyoung started the doctoral program in Textile Technology Management at North Carolina State University in Raleigh, North Carolina. She fully enjoyed extending her knowledge in new fields such as management and industrial engineering as well as textile technology. During her Ph.D. program at NCSU, Eunkyoung was inducted into the Phi Kappa Phi Honor Society. She worked as a research assistant for the NTC Project "Emerging Supply Chain Configurations". Her studies were also supported by the Korean Government.

Eunkyoung intends to return to South Korea after completion upon her Ph.D program. She is eager to apply her knowledge to industry as a consultant and really hopes that her knowledge and experiences gained in the USA will benefit the economy of Korea.

ACKNOWLEDGEMENT

I would like to express my appreciation to Dr. Kristin A. Thoney for her consistent support not only academically but also throughout my time in the USA. It was my first relationship between an advisor and a student in the USA, and it was the best. I learned very valuable things from her, such as respect, patience, creativity, and the balance between work and fun. I really appreciated the numerous hours that she dedicated to understand me and help me conduct research and write this dissertation.

I wish to express sincere gratitude to Dr. Russell E. King for his insight and guidance in completing this work. I am truly grateful to Dr. Thom J. Hodgson for his time and advice to bring this work to a conclusion. I would like to acknowledge Dr. Trevor J. Little for his benevolent attention and help during the research. Their warm and genuine support was appreciated. I also want to thank Dr. David Hinks for his assistance in understanding and building the index coordinates system. Special thanks to Mr. Jeffery D. Krauss, the dyeing and finishing pilot plant manager at the College of Textiles at North Carolina State University, for his help and suggestions while developing the setup matrices.

Gratitude is also extended to committee members of the Textile Technology Paper, Dr. Peter Hauser, Dr. William Oxenham, and Dr. Abdelfattah Seyam, and to those of the Textile Management Paper, Dr. George L. Hodge, Dr. Jeff Joines and Dr. Henry L. W. Nuttle, for their general advice and guidance.

A special word of love and acknowledgement is given to my friends who always are on my side and send me endless loving support despite the distance, from Korea and other states. I will never forget their true friendship. I would like to recognize the people I met in

Raleigh. I sincerely appreciate their time, energy and love they invested to me. Meeting them and the great times I shared with them are the most precious asset I gained in the USA.

Finally, I want to dedicate this work to my parents, my father, Tae-Yong Cho, without whose sacrifice and support I would not be what I am and would not be here, and my mother, Young-Mi Choi, who always believed in me, encouraged me and prayed for me.

TABLE OF CONTENTS

LIST OF TABLES ……..……...…… xi

LIST OF FIGURES ……..……...……xiv

CHAPTER 1. INTRODUCTION ……..… 1

1.1. Coloration Stages within the Supply Chain …….. 1

1.2. Influence of Globalization on Textile Coloration Firms ……..……. 3

1.3. Influence of Quick Response on Textile Coloration Firms …….……. 4

1.4. Overview of Dissertation ……... 5

CHAPTER 2. LITERATURE REVIEW ……..7

2.1. Batchwise Fabric Dyeing ……... 7

2.1.1. Overview ……7

2.1.2. Analysis of the Fabric Dyeing System ……... 9

2.1.3. Production Planning in a Dyehouse ……..14

2.1.4. Scheduling a Dyehouse: Manual vs. Decision Support System ……....16

2.2. General Scheduling Review …… 19

2.2.1. Overview …….. 19

2.2.2. Sequence Dependent Scheduling ……. 20

2.2.2.1. Setup Times ……... 20

2.2.2.2. Single Machine Problems with Sequence Independent Setup Times…….. 22

2.2.2.3. Single Machine Problems with Sequence Dependent Setup Times…….. 25

2.2.2.4. Job Shop Problems …… 28

2.2.3. Dynamic Scheduling ……..…… 29

2.2.3.1. Single Machine Problems …...……... 30

2.2.3.2. Job Shop Problems ……..…... 31

2.2.4. Supply Chain Scheduling …… 34

2.3. Virtual Factory Scheduling Research …….. 37

2.3.1. Overview of Virtual Factory ……..…... 37

2.3.1.1. Scheduling Procedure ……..…... 37

2.3.1.2. Lower Bound …… 38

2.3.1.3. Experimental Results …… 39

2.3.1.4. Hot Job Acceleration …… 40

2.3.2. Sequence Dependent Scheduling …… 41

2.3.2.1. Scheduling Procedures …….. 41

2.3.2.2. Lower Bound …… 42

2.3.2.3. Experimental Results …… 42

2.3.3. Dynamic Scheduling …… 43

2.3.3.1. Scheduling Procedure ……..….. 44

2.3.3.2. Lower Bound …… 45

2.3.3.3. Calculation of *JR* and N_s …… 46

2.3.3.4. Experimental Results …… 46

2.3.4. Supply Chain Scheduling …… 47

2.3.4.1. Scheduling Procedure …… 47

2.3.4.2. Lower Bound …… 48

2.3.4.3. Problem Generation …… 49

2.3.4.4. Experimental Results …… 52

CHAPTER 3. ROLLING HORIZON SCHEDULING OF MULTI-FACTORY SCENARIOS …… 53

3.1. Introduction …… 53

3.2. One Factory …… 54

3.2.1. Problem Generation of Base Case …… 54

3.2.2. Calculation of *JR* and N_s …… 55

3.2.3. Performance Observations of Base Case…… 60

3.2.4. Varying the Total Horizon Length …… 60

3.2.5. Varying the Warm-up Period …… 62

3.3. Two Factories in Series …… 64

3.3.1. Problem Generation of Base Case …… 64

3.3.2. Experimentation of Queuing Schemes …… 65

3.3.3. Performance Observations of Base Case…… 66

3.3.4. Varying the Truck Capacity …… 68

3.3.5. Varying the Total Horizon Length …… 70

3.3.6. Varying the Warm-up Period …… 71

3.4. Other Multi-factory Problems …… 72

3.4.1. Three Factories in Series …… 72

3.4.2. Two Factories Feeding One ……..73

3.4.3. One Factory Feeding Two ……..……74

3.5. Summary and Discussion ……75

CHAPTER 4. CHARACTERIZING SETUPS IN BATCHWISE FABRIC DYEING OPERATIONS ……...…. 78

4.1. Introduction ……...…….…. 78

4.2. Family Formulating and Indexing …….…. 80

4.3. Setup Matrix Structure by Index ……..…... 83

4.3.1. Setup Matrix by Fabric/dye Type Index ……..…… 83

4.3.2. Setup Matrix by Hue Index ……..…… 84

4.3.3. Setup Matrix by Lightness Index ……..…... 86

4.3.4. Setup Matrix by Chroma Index ……..... 88

4.3.5. Interdependency between Setup Matrices ……..……..... 89

4.4. Model Validation ……...…….……... 100

CHAPTER 5. SCHEDULING BATCHWISE FABRIC DYEING OPERATIONS ……..……...…….... 102

5.1. Introduction ……...…….……..…. 102

5.2. Modified Scheduling Algorithm (Algorithm 1**)**……..……103

5.2.1. Shortest Setup Time for Dyeing Operation (SST_D) ……..……..… 104

5.2.2. Performance Evaluation ……..……110

5.3. Experimental Framework ……..……..…111

5.3.1. Processing and Setup Time Estimation ……..…... 111

5.3.1.1. Dyeing Machine ……..…….……... 111

5.3.1.2. Preparation Machine ……..……..…..……..… 114

5.3.1.3. Finishing Machine ……..……..…..……..… 115

5.3.2. Index Coordinates Generation …….…... 116

5.3.3. Experimental Design ……..…….…... 117

5.4. Experimental Results **w**ith Algorithm 1 ……..…. 119

5.4.1. Initial Results …….……..... 120

5.4.2. Algorithm 1 Modifications ……..... 122

5.4.3. Performance of Algorithm 1 Modifications on Problems Generated by Method 2 ..……..…….…….. 127

5.5. Experimental Results **w**ith Policy 2 ……..…… 128

5.5.1. Initial Results …….……..... 128

5.5.2. Performance of Policy 2 Modifications on Problems Generated by Method 2 ……..……...…….……..... 132

5.6. Experimental Results **w**ith Algorithm 3 ……..…..…… 135

5.6.1. Initial Results …….……..... 135

5.6.2. The Effect of Initial γ and τ on the Performance of Algorithm 3 ……..... 137

5.6.3. Performance Comparison to Lower Bound Approximations…..……..….. 140

CHAPTER 6. CONCLUSIONS AND FUTURE RESEARCH …….. 145

6.1. Conclusions ……....145

6.2. Future Research ……..... 146

REFERENCES …….……..……..…....……..... 149

APPENDIX A. REGRESSION CURVE USED FOR EXPERIMENTS ……..156

APPENDIX B. DYEING PROFILE OF EACH FABRIC AND CORRESPONDING DYE GROUP ……..….163

LIST OF TABLES

Table 1.1. Color application methods for apparel [Fralix, 2000] ..……. 1

Table 2.1. Summary of cellulosic and wool fabric dyeing ..……..…..… 7

Table 2.2. The responsibility of departments ……. 10

Table 2.3. The elements of production planning and control ……..… 15

Table 2.4. A list of manufacturing scheduling tools……19

Table 2.5. A setup time matrix [Taner, 2001] ……..…… 25

Table 2.6. The rules compared in Ovacik and Uzsoy's study ……...……. 33

Table 3.1. The parameters for the base case ……..……..54

Table 3.2. *JR* and empirical N_s value for steady state in *One Factory* ……59

Table 3.3. *JR* and empirical N_s value for steady state in *Two Factories in Series*……...65

Table 3.4. *JR* and empirical N_s value for steady state in *Three Factories in Series*…… 72

Table 3.5. *JR* and empirical N_s value for steady state in *Two Factories Feeding One*……73

Table 3.6. *JR* and empirical N_s value for steady state in *One Factory Feeding Two* ……. 75

Table 3.7. The maximum value of the difference in L_{max} - *LB* in days with 2 trucks ……76

Table 3.8. The relationship between simulation parameters and the difference in L_{max} - *LB* ……..... 76

Table 4.1. Eight hue indices in *L* C* h* space ……... 81

Table 4.2. Ten lightness indices in *L* C* h* space ……..... 82

Table 4.3. Five chroma indices in *L* C* h* space ……..... 82

Table 5.1. An example of the job counting procedure ……..107

Table 5.2. Processing times of each fabric type and corresponding dye type……... 112

Table 5.3. Preparation of polyester fabric for dyeing [ICI, 1964] ……..... 114

Table 5.4. Processing time data of some commercial antistatic agents [Pajgrt, O. and Reichstadter, B, 1979] 116

Table 5.5. Comparison of performance and computational time of Algorithm 1 and 2 under due date generation Method 1 120

Table 5.6. Comparison of performance and computational time of Algorithm 1 and 2 under due date generation Method 2 121

Table 5.7. The results of Algorithm 1 modifications 127

Table 5.8. Comparison of performance and computational time of Algorithm 2 and M5 with due dates generated by Method 2 128

Table 5.9. Comparison of performance and computational time of Policy 2 and Algorithm 2 under due dates generation Method 1 with $T = 150$, $\beta = 3$, $\gamma = 0$, $\tau = 150$ 129

Table 5.10. Comparison of performance and computational time of Policy 2 and Algorithm 2 under due dates generation Method 2 with $T = 150$, $\beta = 3$, $\gamma = 0$, $\tau = 150$ 130

Table 5.11. Comparison of performance and computational time of Policy 2 and Algorithm 2 under due date generation Method 1 with $T = 3200$, $\beta = 3$, $\gamma = 400$, $\tau = 4800$ 131

Table 5.11. Comparison of performance and computational time of Policy 2 and Algorithm 2 under due date generation Method 2 with $T = 3200$, $\beta = 3$, $\gamma = 400$, $\tau = 4800$ 132

Table 5.13. The results of Policy 2 modifications 134

Table 5.14. Comparison of performance and computational time of Algorithm 3 and Algorithm 2 under due date generation Method 1 135

Table 5.15. Comparison of performance and computational time of Algorithm 3 and Algorithm 2 under due date generation Method 2 136

Table 5.16. The pattern of γ and τ effect 137

Table 5.17. Relationship between range multiplier and due date range for 20 jobs 142

Table 5.18. Relationship between range multiplier and due date range for 50 jobs 143

Table 5.19. Relationship between range multiplier and due date range for 100 jobs ….. 143

LIST OF FIGURES

Figure 1.1. Textile pipeline and fabric dyeing ……..……..…. 2

Figure 2.1. Key areas in fabric dyeing technology..……. 9

Figure 2.2. Batch dyeing operations flowchart [Koksal et al., 1990] …….…… 11

Figure 2.3. Data flow diagram [Koksal et al., 1990]…… 12

Figure 2.4. Batch dyeing Entity-Relation diagram [Koksal et al., 1990]…….…… 13

Figure 2.5. The benefits of reduced setup time [Yang and Deane, 1993]…….…….. 15

Figure 2.6. Combining batches *i* and *k* [Baker, 1999]……. 23

Figure 2.7. Splitting the last job from batch *i* [Baker, 1999]……..… 24

Figure 2.8. Two consecutive rotations [Taner, 2001]……...… 26

Figure 2.9. Forward insertion of job *i* [Taner, 2001]……27

Figure 2.10. Backward insertion of job *i* [Taner, 2001]……...… 27

Figure 2.11. Scenario generated for *Two Factories in Series* [Thoney et al., 2002a]……. 49

Figure 2.12. Scenario generated for *Three Factories in Series* [Thoney et al., 2002a] ……49

Figure 2.13. Scenario generated for *Two Factories Feeding One* [Thoney et al., 2002a] ...50

Figure 2.14. Scenario generated for *One Factory Feeding Two* [Thoney et al., 2002a] …..51

Figure 3.1. The number of jobs in the system with *JR* of 128 and N_s of 576 ……. 56

Figure 3.2. The number of jobs in the system with *JR* of 126 and N_s of 1 ……. 57

Figure 3.3. The number of jobs in the system with *JR* of 126 and N_s of 378…….…….. 58

Figure 3.4. The regression curve for the empirical N_s value ……... 59

Figure 3.5. Performance of the base case of *One Factory*…….. 60

Figure 3.6. Performance with varying total horizon length in *One Factory* ……..……. 62

Figure 3.7. Performance with varying warm-up period in *One Factory*……..…… 63

Figure 3.8. 95% confidence interval of w = 20 days and 95% workload in *One Factory* .. 63

Figure 3.9. 95% confidence interval of w = 40 days and 95% workload in *One Factory* .. 64

Figure 3.10. Performance comparison between queuing schemes …..……..……. 65

Figure 3.11. Performance of the base case of *Two Factories in Series* ……..…..…….. 67

Figure 3.12. Actual *LB* and L_{max} values for base case of *Two Factories in Series* ....…….. 67

Figure 3.13. Performance with varying truck capacity at 80% workload in *Two Factories in Series* …….. 69

Figure 3.14. Performance with varying total horizon length in *Two Factories in Series* 70

Figure 3.15. Performance with varying warm-up period at 90% workload in *Two Factories in Series* ……. 71

Figure 3.16. Performance of the base case of *Three Factories in Series* ……...……..73

Figure 3.17. Performance of the base case of *Two Factories Feeding One* ……....……... 74

Figure 3.18. Performance of the base case of *One Factory Feeding Two*..…… 75

Figure 4.1. Functional decomposition of batchwise dyeing operation …… 78

Figure 4.2. The top view of the color space given by chroma and hue angle ……. 80

Figure 4.3. Three indices of a job in the color space …….…….. 83

Figure 4.4. Fabric/dye type setup matrix ……... 84

Figure 4.5. Hue setup matrix ……... 85

Figure 4.6. Hue space …….……. 86

Figure 4.7. Lightness setup matrix …….…… 87

Figure 4.8. Lightness space …… 88

Figure 4.9. Chroma setup matrix …… 88

Figure 4.10. Chroma space …….. 89

Figure 4.11. Setup-free zone induced by hue and lightness index …….. 90

Figure 4.12. Expanded setup matrix induced by the hue and lightness indices …….. 92

Figure 4.13. Setup-free zone modification decreased by *Cutoff* (*C*) ……..…….. 94

Figure 4.14. Expanded pink and violet blocks of setup matrix in Figure 4.12 ……95

Figure 4.15. Setup-free zone induced by hue, lightness and chroma …….……. 97

Figure 4.16. Expanded green and yellow blocks of setup matrix in Figure 4.12 ….……... 99

Figure 4.17. Munsell Color System [Burnham, R. W. et. al., 1963] ……..……100

Figure 5.1. The decision flow of SST_D rule ….…….106

Figure 5.2. Dyeing polyester with disperse dyes [Kulkarni et al.,1986]…….……113

Figure 5.3. Flow shop setting.…… 117

Figure 5.4. Graphic illustration of the meaning of initial τ…… 123

Figure 5.5. Graphic illustration of the condition added into step 4.1. ……124

Figure 5.6. Graphic illustration of a bad decision by Algorithm 1 …….124

Figure 5.7. The average L_{max} with varying *Counter* values …….……...126

Figure 5.8. The average computational time with varying *Counter* values …….……...126

Figure 5.9. Strong τ effect and weak γ effect ……..138

Figure 5.10. Strong γ effect and weak τ effect …….……...139

Figure 5.11. Strong γ effect and strong τ effect …….…….…… 139

Figure 5.12. Performance of Algorithm 3 to *LB* approximation with 20 jobs ……142

Figure 5.13. Performance of Algorithm 3 to *LB* approximation with 50 jobs ……142

Figure 5.14. Performance of Algorithm 3 to *LB* approximation with 1-0 jobs …… 143

Figure A.1. Regression Curve from Excel 2000 …… 156

Figure A.2. The number of jobs in the system with *JR* of 102 (80 % workload) and N_s of 159 ……. 157

Figure A.3. The number of jobs in the system with *JR* of 115 (90 % workload) and N_s of 232 ……...……. 157

Figure A.4. The number of jobs in the system with *JR* of 122 (95 % workload) and N_s of 284 ……..157

Figure A.5. Regression curve from Origin 7.0 ..…… 158

Figure A.6. The number of jobs in the system with *JR* of 102 (80 % workload) and N_s of 112 ……..159

Figure A.7. The number of jobs in the system with *JR* of 115 (90 % workload) and N_s of 195 ……..…... 159

Figure A.8. The number of jobs in the system with *JR* of 122 (95 % workload) and N_s of 301 ……..159

Figure A.9. The number of jobs in the system with *JR* of 102 (80 % workload) and N_s of 124 ……...……...160

Figure A.10. The number of jobs in the system with *JR* of 115 (90 % workload) and N_s of 183 …….. 160

Figure A.11. The number of jobs in the system with *JR* of 122 (95 % workload) and N_s of 274 ……. 161

Figure A.12. Performance of the base case of *One Factory* using initial values at which system stabilized ……..….. 162

Figure B.1. Dyeing bleached mercerized cotton with direct dyes [Kulkarni et al.,1986] . 163

Figure B.2. Dyeing bleached mercerized cotton with reactive dyes [Kulkarni et al.,1986] ……...…….……..163

Figure B.3. Dyeing wool with acid dyes [Kulkarni et al.,1986] ……..……..164

Figure B.4. Dyeing rayon with direct dyes [Kulkarni et al.,1986] …..……. 164

Figure B.5. Dyeing acrylic with basic dyes [Kulkarni et al.,1986] …..……. 165

Figure B.6. Dyeing polyamide with acid dyes [Kulkarni et al.,1986] ……. 165

Figure B.7. Dyeing polyester/cotton blend with disperse/reactive dyes

[Kulkarni et al.,1986] …..…….. 166

CHAPTER 1. INTRODUCTION

1.1. Coloration Stages within the Supply Chain

With the exception of certain industrial uses, most textiles are subjected to some forms of the coloration process, either dyeing or printing, during the conversion of fibers to yarns and fabrics. The coloration process can be carried out at several stages as is shown in Table 1.1 [Fralix, 2000].

Table 1.1. Color application methods for apparel [Fralix, 2000]

Coloration method	Applied to	Subsequent Processes
Stock-dyed	Fiber	Spin, Weave, Cut, Sew, Wet process, Press
Yarn-dyed	Sliver	Spin, Weave, Cut, Sew, Wet process, Press
Yarn-dyed	Yarn	Weave, Cut, Sew, Wet process, Press
Piece-dyed	Fabric	Cut, Sew, Wet process, Press
Printed	Fabric	Cut, Sew, Wet process, Press
Belt screen Printed	Cut garment parts	Sew, Wet process, Press
Screen Printed	Finished Garments	Wet process, Press
Garment dyed	Finished Garments	Press

The coloration stage selected is determined by many factors. According to Clarke [1982], economics and fashion are the most important ones. From the perspective of economics, dyeing early in the manufacturing sequence is the least costly. On the other hand, dyeing late can be advantageous due to the elimination of colored waste and a reduction in the time spent holding colored goods inventory. From the perspective of fashion, methods such as stock dyeing and yarn dyeing are slow to respond to changes in consumer preferences because the lead time from the colored fiber to the end product takes several

months. In contrast, fabric coloration can allow fast response to changes in two to three weeks from greige (undyed) fabric [Madaras et al., 1993]. To meet the customers' needs, it is clearly advantageous for the garment manufacturer to suspend the decision of coloration as long as possible, and fabric dyeing is the predominant choice (Figure 1.1). Fralix [2000] reported that about 70% of fabrics are dyed, 20% are printed, 5% are made from dyed yarn and 5% are produced white.

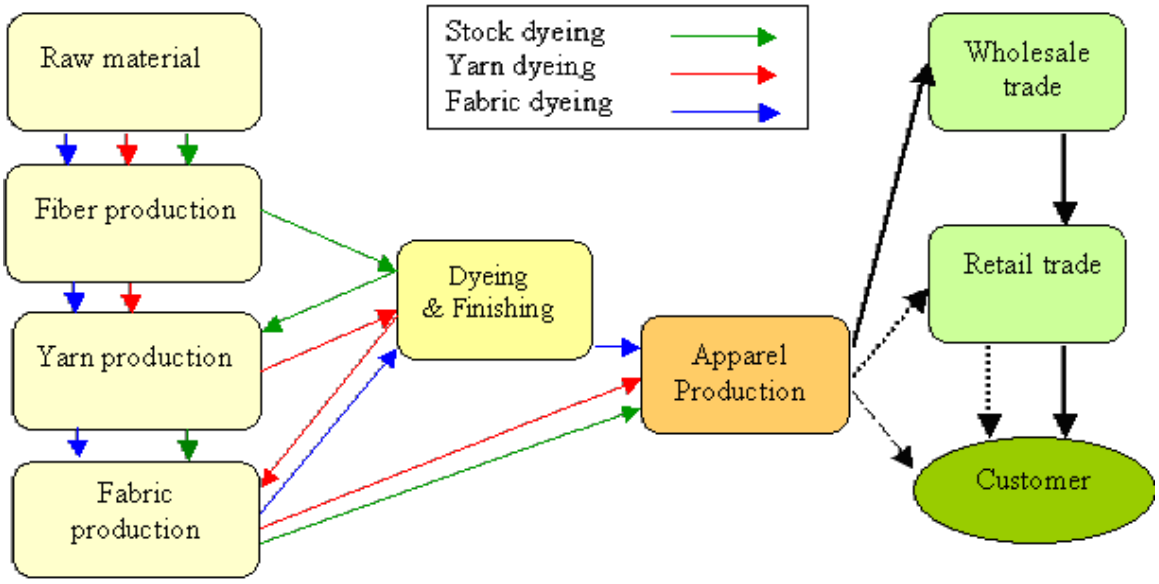


Figure 1.1. Textile pipeline and fabric dyeing

For the fabric dyeing process, the increasing cost of energy and labor over the past several decades has forced the textile industry to consider continuous processes over batch processes [Mock, 1997]. However, for short runs, batch dyeing is generally more economical because of the higher startup and changeover costs associated with continuous dyeing. The advantages such as the versatility, ease of control, and short run capability make batch processes applicable for coloration of many textile products [Kulkarni et. al, 1986]. Now

customer demand for colored fabric is fluctuating and ever-changing. Versatile batchwise fabric dyeing is often favored because it is shorter in time, more flexible and more responsive to customer needs.

1.2. Influence of Globalization on Textile Coloration Firms

Product quality is no longer a new concern in coloration technology but a basic requirement. According to Burdett et al. [1999], the primary issues for the 21st century have focused on the environment and economics. Economics is a substantial concern of developed countries because they have a competitive disadvantage when compared with developing countries with respect to lower labor costs. Because companies are competing increasingly on a global basis, it is critical to utilize labor and other resources in the most effective manner.

Textiles coloration firms' short-term profits are subject to the constraints of available dyeing machines, fabrics, technical labor skill and customer demand for specific dye-lot mixes of stock-keeping units. The use of process optimization techniques and automation helps to solve problems from these constraints [Hoffmann, 1998]. Long-term survival, however, depends on the ability to simultaneously maximize machine productivity, to minimize fabric roll inventory and handling costs, to maximize labor productivity, and to provide optimum production flexibility for customer service. Today, processes for workflow scheduling and utilization of expensive equipment are as important as the more classical quality issues of fastness properties, levelness and color tolerance (i.e. shade match to standard and lot to lot variation).

1.3. Influence of Quick Response on Textile Coloration Firms

In order to respond fast enough in today's highly competitive global markets, effective control of material movement through textile coloration firms is becoming increasingly important. Among different shop performance criteria, machine scheduling problems with a lateness penalty have received tremendous attention. According to Tan et al. [2000], although meeting due dates (on-time delivery) is the most important scheduling objective, only 58% of schedulers across all industries manage to deliver on time. Since most supplier evaluation programs attach considerable significance to delivery dependability, a small improvement in delivery performance might create a substantial competitive advantage. Thus, manufacturers who fail to recognize the importance of scheduling to meet due date requirements and who do not respond adequately to these challenges will be at a competitive disadvantage.

According to Kim and Bobrowski [1994], setup is one of the factors which have significant impact on shop performance. Setup time constitutes a part of flow time that directly affects the throughput rate of a production system. The cost of unit setup time is usually higher than that of unit processing time. A setup change is often made only by a qualified technician. Numerous cases have been reported about the successful reduction of setup times by the standardization of setup procedures. However, setup times have not been eliminated and remain an important element of real problems in production systems. Setup times are especially critical when they are sequence dependent like those in coloration operations. A sequence dependent setup implies that the setup time of a job is a function of the preceding job on the machine and therefore the overall sequence. Panwalkar et al. [1973]

discovered that about three quarters of managers reported that at least some operations they scheduled required sequence-dependent setup times, while approximately 15% reported that all operations required sequence-dependent setup time.

1.4. Overview of Dissertation

For long-term survival, the fiber-textile-apparel-retail chain must challenge itself to become more responsive to customer needs, delivering products on time and maintaining the highest standards [Cooper et al., 1998]. The goal of the dissertation is to develop approaches to improve the performance of the fiber-textile-apparel-retail chain by combining several research ideas. Efficient production scheduling is an essential part of supply chain management in industry. Thus, one way to improve the performance of the fiber-textile-apparel-retail chain is to optimize the performance by scheduling the entire organization simultaneously and not just the individual entities separately. Furthermore, the performance of each entity could be improved by examining the specific characteristics of each entity and refining the scheduling accordingly.

Previous research at NCSU led to the development of a scheduling system called the Virtual Factory. The Virtual Factory has been found to provide near-optimal solutions to industrial-sized problems with the objective of meeting customer due dates. Research with the Virtual Factory has produced a methodology for scheduling transportation and thus enabled the scheduling of multi-factory supply chains. Further Virtual Factory research resulted in a rolling horizon procedure for single factories that more accurately simulates the conditions under which a scheduling system would be used in industry. Therefore, the first

goal of this dissertation is to test the rolling horizon implementation of the Virtual Factory in multi-factory scenarios.

Among the entities in the fiber-textile-apparel-retail chain, batchwise fabric dyeing operations need to be carefully studied because they have significant sequence dependent setup times. Recent research with the Virtual Factory has produced a methodology for scheduling machines with simplified sequence-dependent setups. This provides an initial step in incorporating batchwise fabric dyeing operations into the Virtual Factory. Consequently, the second goal of this dissertation is to develop a scheduling algorithm for dyeing operations.

In Chapter 2, the literature review is presented. Chapter 3 explains the results of testing the performance of the rolling horizon implementation of the Virtual Factory for multi-factory scenarios. In Chapter 4, setups in batchwise fabric dyeing operations are characterized. Chapter 5 addresses the development of a scheduling algorithm for dyeing operations and presents experimental results. Finally, Chapter 6 concludes this research and discusses future research.

CHAPTER 2. LITERATURE REVIEW

The literature review has been divided into 3 parts. First, Section 2.1 discusses the batchwise fabric dyeing system and the current methods of production planning and scheduling in a dyehouse. Then, Section 2.2 provides an overview of topics in the scheduling literature related to scheduling dye operations in the textile and apparel supply chain. Finally, Section 2.3 explains the Virtual Factory and its development in relation to the topics described in Section 2.2.

2.1. Batchwise Fabric Dyeing

2.1.1. Overview

The general sequence of batchwise fabric dyeing is fabric preparation, exhaustion, and after-treatment. Dyeing processes and the corresponding dyes vary with fiber type (Table 2.1).

Table 2.1. Summary of cellulosic and wool fabric dyeing

	Cellulosic fabric [Madaras et al., 1993]	Wool fabric [Bearpark et al., 1986]
Preparation	Singeing, Desizing, Scouring, Washing, Bleaching, Mercerizing	Carbonizing, Scouring, Milling, Setting, Drying, Raising, Bleaching
Dyes	Surfur dyes Vat dyes Azoic (Naphtol) dyes Direct dyes Reactive dyes	Level-dyeing acid dyes Milling acid dyes 1:1 metal complex dyes 1:2 metal complex dyes Chrome dyes Reactive dyes
Machinery	Rope form (Winches, Jets, Overflow machines, Short-liquor machines) Open-wide (Jigs, Perforated beams)	Winches, Jets, Beam dyeing machines

Holme [2001] summarized recent trends in fabric dyes and dyeing technology. Cotton and other cellulosic fibers are the predominant textile fabrics produced today. Environmental pressure is on sulfur and vat dyes because of their subsequent chemical reduction process. Moreover, the fashion movement out of blue denim has also had consequences for synthetic indigo manufacture. Azoic dyes are now less attractive because of increased fears of carcinogenicity. Direct dyes are still in use, but the use of reactive dyes has grown with decreasing prices. Wool is still important, although it constitutes just 4% of world textile fiber consumption. The use of acid levelling, acid milling types and 1:1 metal complexes is slowly being replaced by the use of 1:2 metal complex and also by reactive dyes. The man-made fiber with highest consumption is polyester, and it is very widely used in polyester/cellulosic fiber blends. Disperse dyes for polyester application have been actively researched, and this dye class is, by far, the most important for polyester. For fabric dyeing machines, winch (beck) dyeing machines are in decline, and many manufacturers have produced different forms of jet dyeing machines.

The demands of fabrics from batch dyeing processes are for casualwear, leisurewear and sportswear. Therefore, satisfactory levels of colorfastness to washing, rubbing and light, combined with attractive aesthetics, are required [Burdett at al., 1999]. A high stage of technical development in every component section is necessary to meet these requirements (Figure 2.1). Developments in dyestuff chemistry have emphasized high levels of batch to batch color reproducibility and higher colorfastness. Many developments in dye chemistry and dyeing technology have concentrated on the accumulated knowledge of the physical chemistry of the dyeing process. Understanding of color physics, such as color measurement, color matching, and color control, has been running parallel with advances in automated

control. The importance of color communication has been realized, and related research has increased. Automated gravimetric weighing and dispensing systems provide precise recipe reproducibility. Modifications of machines make it possible to process in reduced cycle times using less water, dyes chemicals and energy. An automatically controlled plant provides more information of higher quality on process conditions, leading to potential improvements in process operation. However, relatively little research is available in the area of dyeing operations from a scheduling perspective.

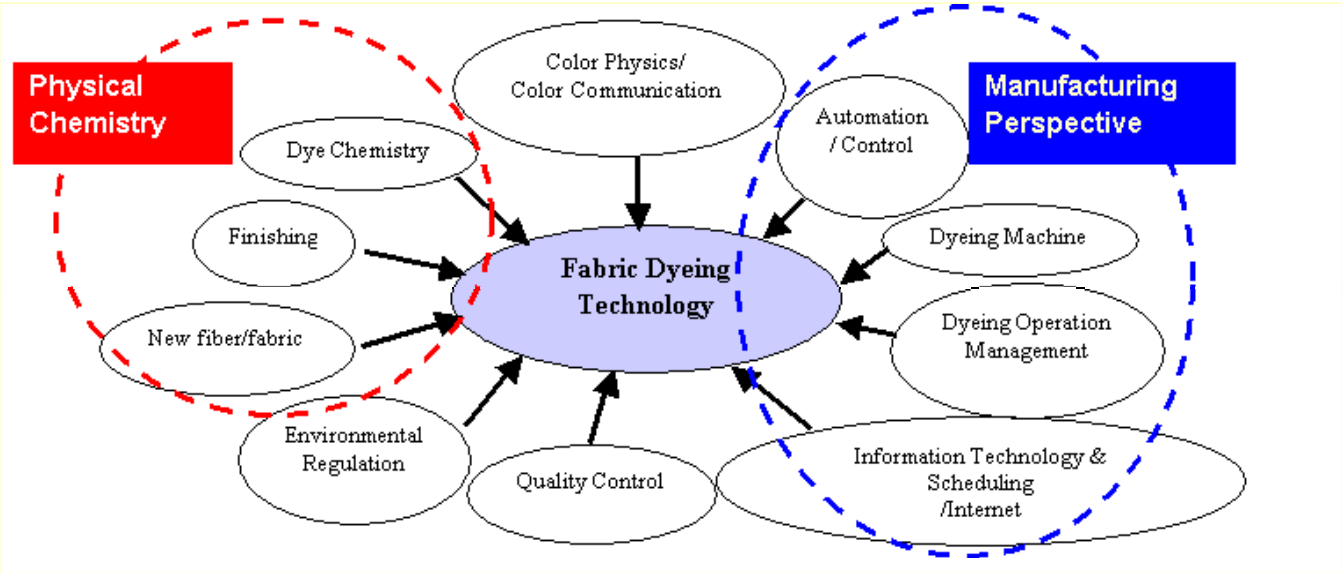


Figure 2.1. Key areas in fabric dyeing technology

2.1.2. Analysis of the Fabric Dyeing System

To understand the basic activities of fabric dyeing and their relationships in the process, the dyeing system was investigated. Modern textiles dyeing operations have been well analyzed through structured system analysis [Koksal et al., 1990, 1992, and 1998]. Table 2.2 summaries the primary responsibility of the different departments in order to carry out the basic activities by which dyed goods are produced.

Table 2.2. The responsibility of departments

Department	Responsibility
Shade library	Interpretation of customer orders by comparison to shade library
Dyelab	Determination of all specifications regarding the substrate; the color formula and chemicals; the process type
Engineering and Planning	Process specifications in terms of - procedures : the machines / amount of materials necessary / processing steps / parameter protocol / setting - schedules : the sequence of jobs / machine assignment /due dates
Greige area	Weighing the greige fabric, cutting into pieces and putting into lots
Drug room	Preparation of solutions for the dyebath
Materials supply room	Support the dyehouse and drug room with dyes and chemicals
Utilities	Support the dyehouse and drug room with water, air, electricity
Dyehouse	Actual dyeing processes on fabric
Control room	Control of the procedures by means of computers, controllers, monitors, and audio systems.
Shade room	Quality check of processed fabric samples
Customer service	Release of the fabric by a shipping order

Dyehouse operations, including the storage, inspection activities, delays and decision points, can be seen in the material flow diagram in Figure 2.2. Although material flow among departments is usually obvious, that is not always the case for information. The data flow diagram in Figure 2.3 was suggested in order to detect inefficient flows and missing, redundant or irrelevant data.

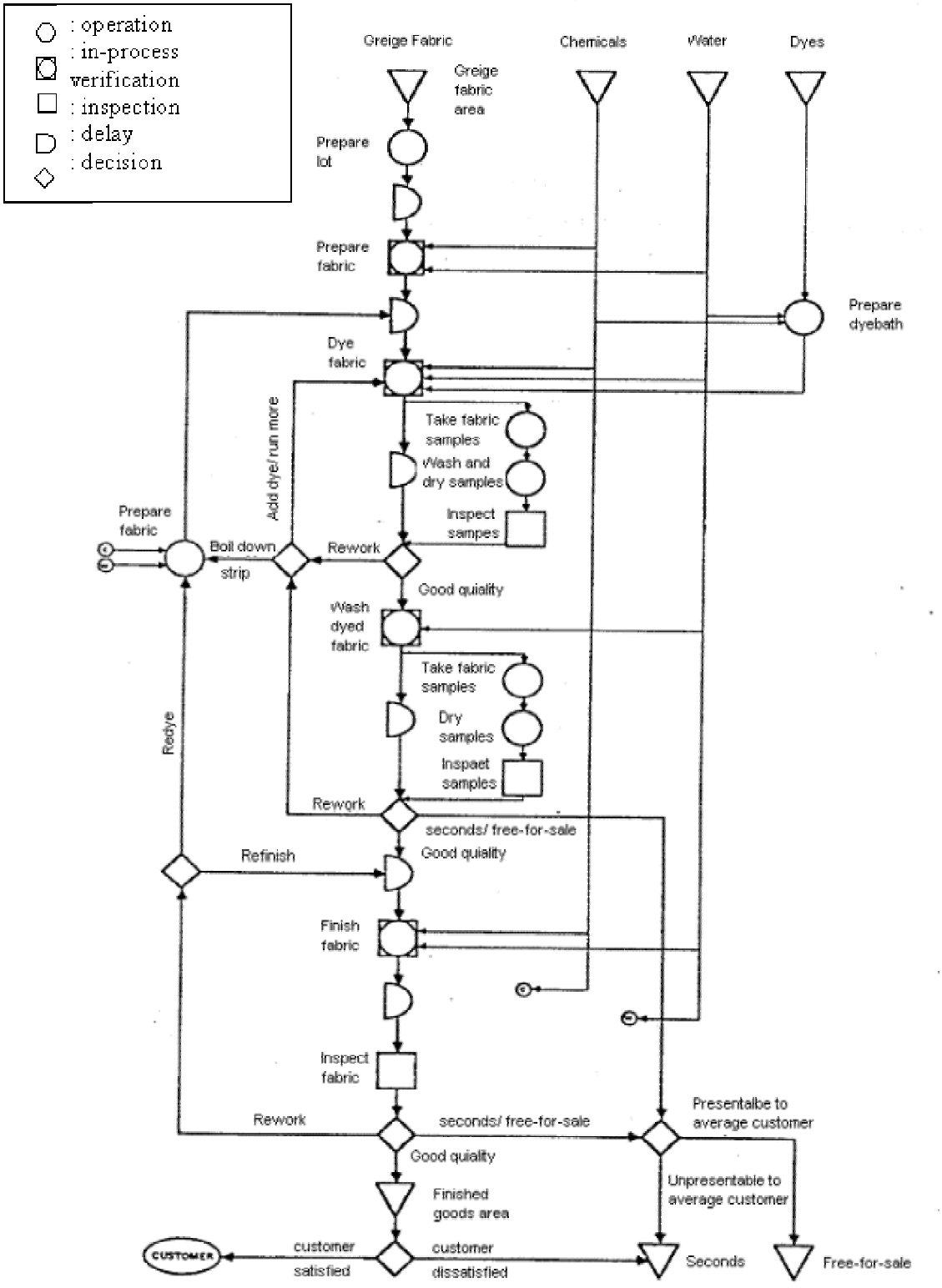


Figure 2.2. Batch dyeing operations flowchart [Koksal et al., 1990]

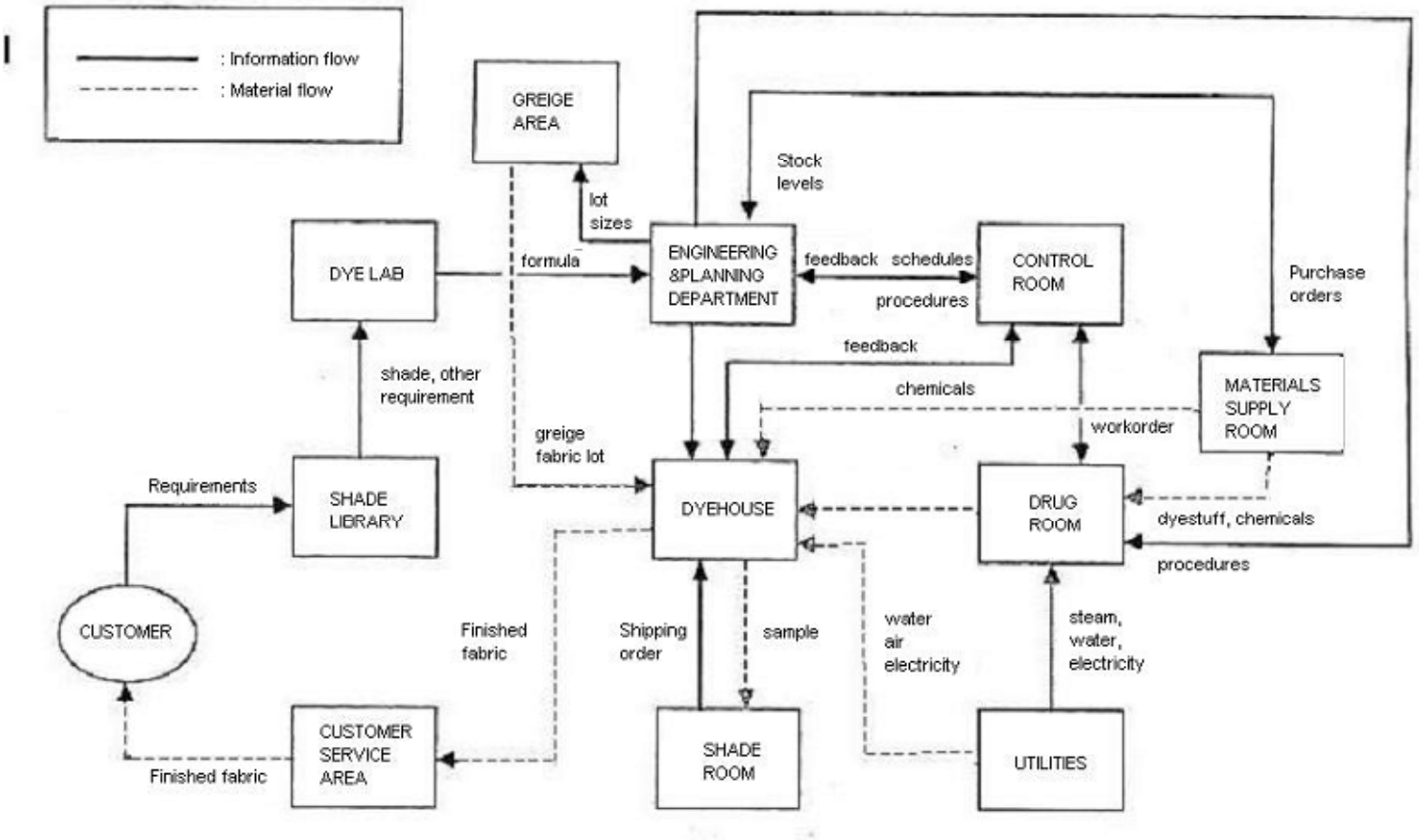


Figure 2.3. Data flow diagram [Koksal et al., 1990]

Another structured system analysis tool useful in dyeing operations is the entity-relationship diagram. The relationships between factors critical to the dyeing process, including materials, logistics, equipment, controls, processes, and human factors, is depicted in Figure 2.4. Koksal et al [1992] assured that these tools resulted in information which could lead to better planning and control in dyeing system, improved quality, reduced costs and shorter time to market.

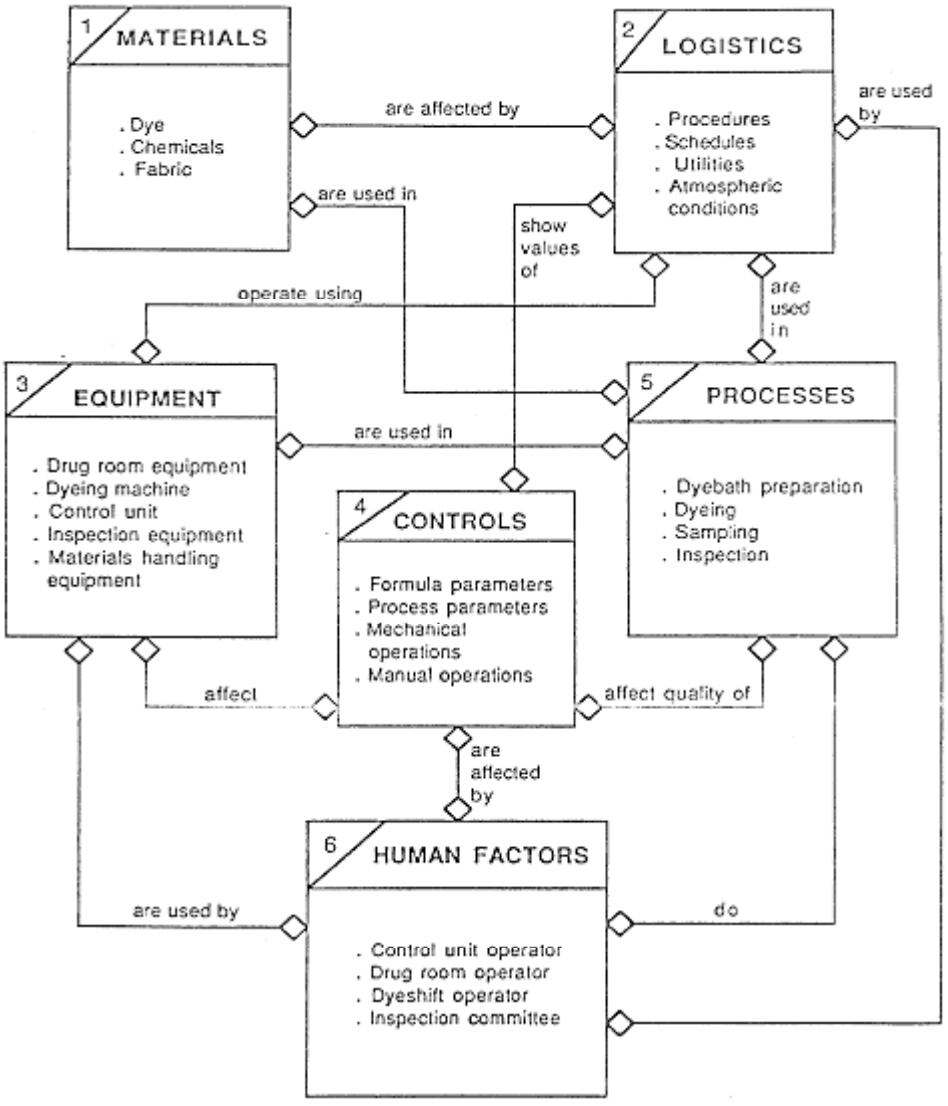


Figure 2.4. Batch dyeing Entity-Relation diagram [Koksal et al., 1990]

2.1.3. Production Planning in a Dyehouse

In the analysis by Nanda [1998], the factors that cause traditional dyehouse problems are the tendency towards the large number of variables (yarns, fibers, fabrics, dyes and finished) and seasonal levels of activity. The first factor can be significantly improved by the optimization of the dyeing process. Traditionally, most of the research published on dyeing has been on achievements in optimizing the mechanical and chemical technology associated with the physical dyeing process. These advances are closely related to production control, in that products are made to the desired quality by the best or most cost effective methods. The second factor makes planning, the maximum use of capacity and the optimum use of labor extremely important, but also difficult. Production planning is more specifically concerned with determining how to employ production resources over a given future time period in response to the ordered or forecasted demand for the product. It essentially consists of five interlinked elements (Table 2.3).

Scheduling is an important part of production planning, especially regarding the reduction of cycle time which enhances the agility and flexibility of a firm. Schedulers should take into account that setup times play a critical role in the performance of the machine shop, since reduced setup time directly leads to shorter cycle times. The relationship between setup time reduction and performance improvement in a stochastic closed manufacturing cell was addressed by Yang and Deane in 1993 (Figure 2.5).

Table 2.3. The elements of production planning and control

Element	Meaning and Example
Routing	The route path of a manufacturing lot through the dyehouse e.g.) For a polyester/ cotton blend dyeing Heat setting – Disperse dyeing – Scutching – Drying - Reactive dyeing – Soaping – Drying
Loading	Loading of work against the concerned machine or a worker e.g.) Decide how many stenters to employ for each job; drying or heat setting.
Scheduling	Determining starting and completion time for various operations Categorizing the jobs according to their preferential value. e.g.) Dye a pastel first before a dark shade, if the same dyeing machine is used.
Dispatching	Releasing orders at predetermined timings e.g.) Bring an empty trolley to the dyeing machine for unloading the dyed lot before process completion to minimize the machine idle time
Expediting / Follow-up	Ensuring that the materials, tools and equipment are received by the worker in right time for various operations

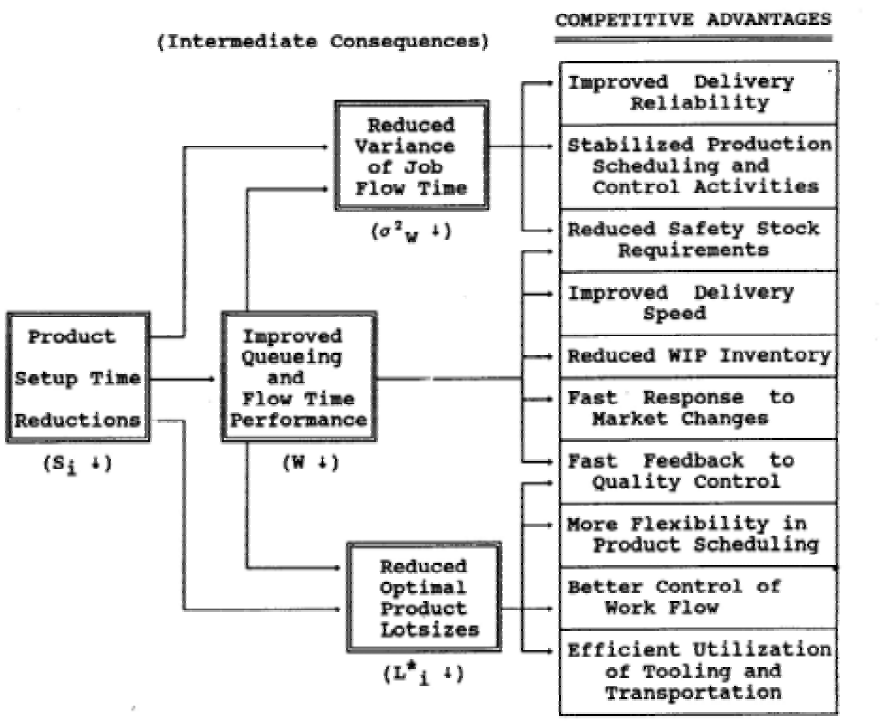


Figure 2.5. The benefits of reduced setup time [Yang and Deane, 1993]

Morales et al. [1996] and Maldonado et al. [2000] investigated the applicability of the optimization model in order to obtain the best color sequence that produced the minimum differences between desired colors and those obtained in the process of fabric dyeing. Traditionally, the choice of sequence was made by technical personnel on the basis of the 'lightness' and 'cleanness' of the color and their own experience. By determining the level of contamination by residual dyestuff and analyzing the quantitative consequences, they showed that their problem was an instance of the asymmetric traveling salesman problem. However, it was not always possible to find a sequence that eliminates all the differences or that met all the imposed restrictions. In that case, the dyeing equipment had to be cleaned and it incurred setup time.

2.1.4. Scheduling a Dyehouse: Manual vs. Decision Support System

The traditional way of scheduling a dyehouse was briefly mentioned by Moor [1997]. It has been a manual activity requiring the scheduler to concurrently consider a number of factors such as:

- When is the order due to the customer
- How to subdivide a large order to achieve the desired results in the time allowed
- How to maintain efficiency
- How to reschedule jobs to accommodate high priority in unplanned events
- What impact will the change have on subsequent processes

Due to changes in technology such as faster computers, more computer memory, and efficient scheduling algorithms, Decision Support System (DSS) for production scheduling can synchronize demands and constraints in manufacturing plants. In the area of dyeing

operation scheduling, however, the approach of a DSS is relatively new. Not many case studies exist.

A case study of DSS implementation, “Asprova” [Nara Lace, Co. Ltd, 2003] showed problems in the traditional management of a dyehouse and the benefits of the DSS approach. Nara Lace, Co. Ltd is a leading manufacturer, primarily in dyeing curtain lace in Japan. The company had been drafting schedules of their dyeing process based on offline data of customer orders and inventory. The schedule was drafted manually on a daily basis. In order to modify the existing schedule, the schedule manager would have to work overtime at night or report to duty early on the following day. Furthermore, drafting and modification of the schedule had to be made by personnel who were familiar with the process, equipment and constraints. This meant that only the schedule manager could improve overall efficiency, but it was nearly impossible due to the complexity of the process and the overwhelming workload. Finally, due to the difficulties in making modifications to the data in the mainframe, rescheduling was done by writing directly on the work instructions. As a result, the schedule manager had to bear the additional responsibility of handling customer inquiries. Lack of visibility of schedule data in the factory resulted in heavy dependence on the schedule manager.

Due to an increase in production and customer orders, the company realized that manual rescheduling was no longer possible. Nara Lace, Co. Ltd. underwent a major restructuring plan to systemize production scheduling and peripheral systems. Master database preparation was a painstaking process. Then, an interface program connecting the peripheral system was developed. The implementation eventually turned out to be a total success, especially in terms of run time. The original manual scheduling system took 3-4

hours to draft a schedule, whereas Asprova could achieve the same result within a half hour.

After implementing Asprova, modifications to the schedule could be made easily, thus reducing unnecessary workload. Besides eliminating the hassle of writing instructions manually, inquiries concerning customer due dates could be handled efficiently. Bottlenecks in the dyeing process were reduced as a result of the increased accuracy in scheduling. Hence, waste at production was minimized. Due to the quicker run time and reduction of hassle in data input, it was possible to evaluate the on-site progress while scheduling and lengthening the deadline for order input. Previously, these could not have been possibly done without the schedule manager. With Asprova, even inexperienced personnel could handle scheduling with ease, since the necessary process specifications had been set up as a master database.

Generalized DSS for scheduling and machine loading in plants were reported, specializing in dyeing piecegoods with multiport machines in a pressure beck [Cooper et al., 1998]. This approach inputs information into mathematical and heuristic models that represent the essential features of the decision rules. It simulated outcomes, measured possible outcomes against competing goals of the decision process and presented them for choice. It was reported that this DSS approach optimized dye machines, material, and labor utilization within the decision rules of the piece-dyeing operations.

Gaston Country Dyeing Machine Co. introduced a dyehouse management system, SuperTex +SQL [Moor, 1997 and Unknown, 1997]. It emphasized the importance of the right information and communication between departments. It was reported that this system provided not only the production data analysis, but also excellent assistance to the scheduling task. However, in these commercial articles, only a general understanding of the DSS and its benefits were reported. The discussion about the model itself was missing.

2.2. General Scheduling Review

2.2.1. Overview

Although production planning and scheduling has not been extensively studied for dye operations, much general information can be found in the industrial engineering literature. A brief history of manufacturing scheduling is shown in Table 2.4. [Gould, 2002].

Table 2.4. A list of manufacturing scheduling tools

Name	Features
Backward scheduling	• Taking an order's due date, subtracting the processing times, and ending with a start date for that customer order.
MRP (Materials Requirements Planning)	• Generating production and purchase orders based on backward scheduling to avoid shortages.
MRP II (Manufacturing Resource Planning)	• Addition of the master production schedule (MPS) on top of MRP. • MPS focused on sales and marketing's demand forecast.
CRP (Capacity Requirements Planning)	• Identifying under-utilization and overload conditions at a machine or work cell.
FCS (Finite Capacity Scheduling)	• Emphasizing maximizing production resources • Making the scheduling problem a mathematical sequencing problem.
OPT (Optimized Production Technique)	• Identifying production bottlenecks • Eliminating them through proper scheduling.
APS (Advanced Planning and Scheduling)	• Batch process and job shop problem can be treated • Overcoming infinite capacity and fixed lead time assumptions of MRP • Estimating potential delays in existing orders. • Rescheduling materials and other resource capacities simultaneously, not sequentially.

Conventional manufacturing planning tools such as MRP help make routine judgments about outsourcing, overtime, new equipment, tooling, labor and other resource issues. However, according to Wortman [1992], these tools rarely account for such operational issues as setup times that are sequence dependent; queue times or other resource

bottlenecks; multiple resource requirements for an operation; and operating or strategic flow and timing constraints such as batching and lot processing rules. This results in excess overtime, delayed production, untimely material receipt, excess inventories, improper scheduling priorities, increased expediting and higher shipping costs.

In this context, research on general scheduling, related to scheduling dye operations in the textile and apparel supply chain, is given. Section 2.2.2 discusses sequence dependent scheduling. Then, Section 2.2.3 provides on literature review on dynamic scheduling. Finally, Section 2.2.4 explains the perspective of supply chain scheduling.

2.2.2. Sequence Dependent Scheduling

2.2.2.1. Setup Times

According to Allahverdi et al. [1999], setup operations have long been ignored, or considered as part of the processing time. However, treating the setup time separately from the processing time allows operations to be performed simultaneously and hence improves performance. This concept is inherent in recent production management philosophies and techniques such as just-in-time (JIT), optimized production technology (OPT), group technology (GT), cellular manufacturing (CM), and time based competition. A key element of the Japanese JIT (kanban/ pull/ stockless) production system, in order to reduce inventories and increase productivity, has been the reduction in preparing time for production. However, Krajewski et al. [1987] showed that using a kanban system did not guarantee improvement in productivity for all systems. Simultaneously reducing setup times and lot sizes was found to be the single most effective way to cut inventory levels and improve customer service. While JIT suggested reducing setup time everywhere, Patterson

[1993] suggested setup time reductions on bottleneck resources by using the scheduling software package OPT.

The difficulty of scheduling dye operation is well known because setup times are sequence dependent [King et al., 2002]. That is why technicians aim to reduce the consequences by first dyeing light colors and then dark. There are numerous industrial systems for which the amount of setup time varies considerably depending on the processing sequence of the jobs [Allahverdi et al. (1999), Das et al. (1995), Franca et al. (1996), Kim and Bobrowski (1994), Srikar and Ghosh (1986)]. The following are examples related to the coloration process.

- The textile industry: Setups for dyeing operations: machine cleaning depends on the color of the current and next order.
- Commercial printing industry: The presses have to be cleaned and settings changed depending on the color of ink, size of paper and types used.
- Plastics manufacturing: In the production of raw and finished plastics of different types and colors, a set of orders constituted by items of different colors has to be assigned to a set of extrusion machines. When the color of those items that were previously processed and presently processed is not same, an amount of plastic is wasted.

In the following sections, research related to scheduling operations with sequence dependent setup times to meet customer due dates is discussed. The criterion of minimizing maximum lateness was chosen as the performance measure to meet this objective. Section 2.2.2.2 provides an overview of research on problems with sequence independent setup times, since many of the approaches used to treat these problems are the foundation of

techniques used to solve problems with sequence dependent setups. Section 2.2.2.3 and 2.2.2.4 present research on sequence dependent setup times for the single machine and job shop problems, respectively.

2.2.2.2. Single Machine Problems with Sequence Independent Setup Times

Monma and Potts [1989] considered situations where a large number of jobs are grouped into a relatively small number of batches. All jobs are available for processing at time zero. A setup time is incurred whenever there is a switch from processing a job in one batch to a job in another batch. Monma and Potts showed that many scheduling problems with batch setup times can be solved in two stages. First, the order of the jobs within each batch is fixed by the rule that ordering the jobs within each batch by earliest due date (EDD) rule gives an optimal schedule for the maximum lateness problem. Then, in the second stage, the ordered batches are merged into an overall schedule using a dynamic programming algorithm. With regards to the computational complexity, Monma and Potts reported that the maximum lateness problem is NP-hard, even with sequence independent setup times. The rule in this procedure was repeatedly used when jobs are ordered within a family [(Baker, 1999) and (Taner, 2001)].

In Baker [1999]'s study, it was assumed that jobs are identified as part of distinct families and that setups are required when the machine changes from processing jobs of one family to another. All jobs are simultaneously available for processing, and the objective is to minimize the maximum lateness. Two heuristic procedures (EDD and GT) were described, and new heuristic procedures (GAP and improvement procedures) were developed.

The Group Technology (GT) sequencing placed the jobs in batches, one batch per

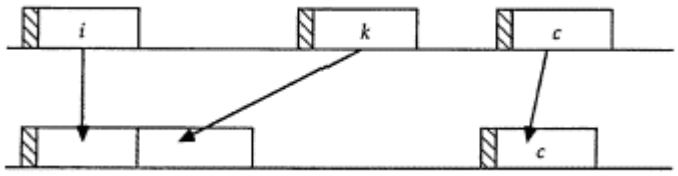
family, and then sequenced the batches from earliest to latest batch due date when setup times were quite large. The GAP heuristic stemmed from the intuition that only if consecutive jobs from the same family have due dates sufficiently far apart, should we consider placing them in separate batches. A simple heuristic was constructed around the Gap condition.

Gap heuristic

```
Step 1 Start with the jobs in EDD order
Step 2 Place them, one at a time, into the sequence.
```

- If its family matches the family of the last batch
  → Add to the last batch.
- If not
  → Add to the latest batch already scheduled for its family
- If such a batch exists & if the Gap condition fails
  → Add the job.
- Otherwise → Add to the end of the sequence, initiating a new batch.

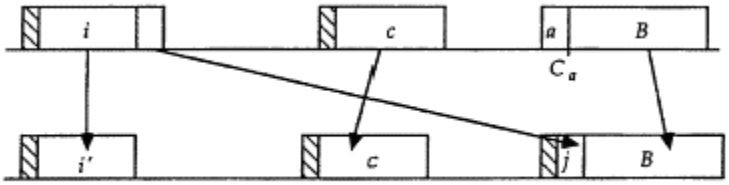
Improvement routines developed by Baker [1999] included two approaches. A *C-neighborhood* is defined as the set of schedules obtained by choosing a batch and combining it with the next earlier batch of the same family. This process starts with the critical batch, *c*. If there is no improvement, the batch considered for combining is shifted to the batch, *k,* that directly precedes the critical batch. This batch is combined with the next earlier batch, *i* (Figure 2.6)



c : the critical batch in which the maximum lateness occurs

Figure 2.6. Combining batches *i* and *k* [Baker, 1999]

An *S-neighborhood* was defined as the set of schedules obtained by splitting off the last job from a batch and moving the job into a later batch corresponding to the same family in the schedule, possibly as a separate batch. Splitting job *j* would result in a new batch and an incremental setup but there is the possibility of moving job *j* into a later batch *B* of the same family. After splitting, the lateness of batches in *B* should be lower than L_c (Figure 2.7).



B: a set of consecutive batches at the end of the schedule with the property $L_B < L_c - s$
a : the last batch in the original schedule prior to the set *B*
C_a : its original completion time.
T : the start time of processing for batch *k* in the original schedule.

Figure 2.7. Splitting the last job from batch *i* [Baker, 1999]

Since the GT sequence might have too few batches, the GT sequence was improved by using the S-neighborhood search (GT/S). Since the Gap heuristic tended to have too many setups, it was logical to improve Gap heuristic using the C-neighborhood search (Gap/C). Finally, some iteration between the two types of neighborhood searches was conceivable (GT/SC and Gap/CS). The performance of the four procedures was compared and overall, Gap/CS produced the best average performance for each problem size [Baker, 1999]. The improvement routines and the best results of Gap/CS are related to Taner [2001]'s heuristic approach on sequence dependent setup times which will be discussed in the next section.

2.2.2.3. Single Machine Problems with Sequence Dependent Setup Times

Taner [2001] considered an *n*-job, single machine scheduling problem with sequence dependent family setup times. It was assumed that the jobs are released all at once. There are N families with n_i jobs in each family i =1, 2, …, N. Setup, constant s, is required only if a job from a smaller indexed family is an immediate successor of one from a larger indexed family. Table 2.5 shows the changeover matrix for family setups.

From/To	1	2	. . .	$N-1$	N
1	0	0	. . .	.	0
2	s	0	. . .	.	0
.	.	.	. . .	.	.
.	.	.	. . .	.	.
.	.	.	. . .	.	.
$N-1$	s	s	. . .	0	0
N	s	s	. . .	s	0

Table 2.5. A setup time matrix [Taner, 2001]

Notation:

d_j - Due date of job j

p_j - Processing time of job j.

F_i - Set of jobs which belongs to family i

F_i^r - rth partition of F_i

R - Number of rotations/ partitions

T - Starting time of the last job of type i (job h) in rotation r

K - Sum of the processing times in the interval between Job h and Job k

First, characteristics of an optimal solution were determined.

Lemma 1: Given the partitions for i = 1, 2,…N and r = 1, 2,…R; sequencing the

jobs in each partition in due-date order minimizes L_{max}.

Lemma 2. There exists at least one optimal schedule that satisfies the following three inequalities for each family $i = 1, 2,\ldots N$ and rotation $r = 1,2,\ldots(R\text{-}1)$ (Figure 2.8).

$$T + K + s + p_h - d_h \geq L' \qquad (2.1)$$

$$L' + p_k \geq T + p_h + K + s + p_k - d_k \qquad (2.2)$$

$$\max_{j \in F_i^r}\{d_j\} \leq \min_{j \in F_i^{r+1}}\{d_j\} \qquad (2.3)$$

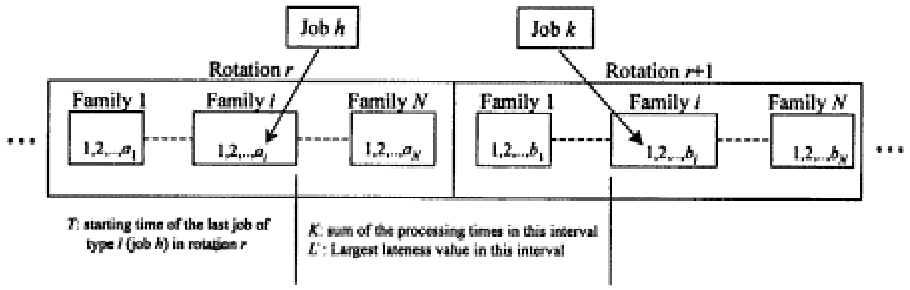


Figure 2.8. Two consecutive rotations [Taner, 2001]

Then, a heuristic solution approach, Algorithm 1, was developed based on these characteristics. The solution technique has three fundamental components:

***Forward insertion*:** The last job of each family within a rotation was considered for insertion before the first job in the corresponding family in the next rotation.

Notation:

I - The job being considered for insertion

J - The latest job between job I's current and potential positions including job I

J' - The latest job between job I's current and potential positions not including job I

K_j - The sum of the processing times of all jobs after job j and before the insertion point of job I

T^J - The total processing and setup time up to and including job J in the current sequence

f_j - Job j's family index.

Corollary 1: Forward insertion of job i (Figure 2.9) should be pursued

if $s + K_J < d_I - d_J$ for $f_J > f_I$

if $K_J < d_I - d_J$ for $f_J \leq f_I$

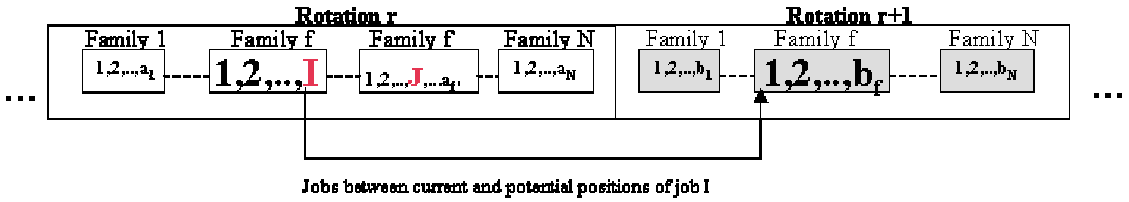


Figure 2.9. Forward insertion of job i [Taner, 2001]

***Backward insertion*:** The first job of each family within a rotation was considered for insertion after the last job in the corresponding family in the preceding rotation

Corollary 2: Backward insertion of job (Figure 2.10) should be pursued

if $L_{J'} + p_J < L_J = L_I$

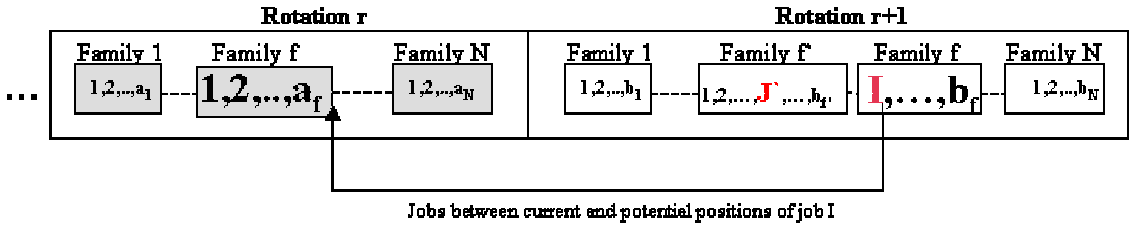


Figure 2.10. Backward insertion of job i [Taner, 2001]

***Setup insertion*:** An additional setup was inserted after the L_{max} job.

Algorithm 1:

Step0 Set r =1 and S_r ={[F_1^1, F_2^1, …, F_N^1]} where jobs within each family are in EDD order. This is the optimal schedule with one rotation.
Step1 Insert a setup immediately following the L_{max} job.
Step2 Set $r = r$ +1 and S_r ={[F_1^1, F_2^1, …, F_N^1] ,…, [F_1^r, F_2^r, …, F_N^r] }
Step3 Consider all forward insertion moves according to Corollary 1. If at least one move has been made, repeat Step 3.
Step4 Consider all backward insertion moves according to Corollary 2. If at least one move has been made, repeat Step 4.
Step5 Consider all forward insertion moves according to Corollary 1. If at least one move has been made, go to Step 3.
Step6 Save S_r and its associated L^r_{max} value. If L^r_{max} comes from a job which was shifted forward in the schedule due to the most recent setup insertion, go back to S_{r-1} . Insert a setup immediately following the L^r_{max} job, set $r = r$ −1 and go to Step 2.
Step7 If the final setup is followed by a vacuous rotation, finished. Otherwise, go to Step1.

Algorithm 1 exhibited a propensity for the procedure to get stuck in a local optimum. An improvement procedure was proposed as a post processing mechanism [Taner, 2001]. The method was evaluated via extensive computational experimentation. The results indicated that the proposed technique was highly effective and runs fast even for large problems such as 10 family, 10,000-job problems.

2.2.2.4. Job Shop Problems

Even for a system in which everything is static and deterministic, optimally scheduling a job shop is known to be hard to solve. Scheduling problems in real world situations are dynamic and full of various uncertainties (machine breakdowns, for example). Past studies show that there has not been a significant amount of research done on scheduling procedures for job shops with sequence dependent setup times [Zhou and Egbulu, 1989]. Broadly speaking, shop time and due date-performance are of primary interest. According to Baker [1984], in actual shops, meeting due-dates tends to be a more important criterion than

minimizing shop time. However, there is no single, universally-accepted measure of effectiveness on this dimension.

According to O'Grady and Harrison [1988], most scheduling problems have been shown to belong to the class of NP-complete problems. Thus, many analytical techniques are infeasible for larger problems due to excessive computation. Consequently, one of the most popular approaches to large job shop problems is to use a dispatching rule to select the next job to be processed from a set of jobs awaiting service.

Kim and Bobrowski [1994] classified and tested four dispatching rules by considering whether sequence dependent setup time and/or due date information was employed. The results demonstrated that there was a significant difference in performance between setup-oriented and ordinary sequence rules. The most significant impact of sequence dependent setup times using ordinary sequencing rules was loss of productivity due to longer flow time and a large increase in production costs due to poor performance in due date related measures. This study also showed that due date information should be included in the sequencing decision in order to provide better due date performance.

According to Zhou and Egbulu [1989], heuristics coupled with simulation is another major approach for job shop problems. A priority function is computed for each job and jobs are selected for processing based on this value. Simulation models of the shop floor are used either to establish the selection of a rule or to evaluate a given set of rules.

2.2.3. Dynamic Scheduling

Up to now, we have dealt only with static problems in which all jobs are released simultaneously at time zero, but jobs in industry are released periodically onto the shop floor.

Therefore, dynamic problems also need to be studied. Section 2.2.3.1 and Section 2.2.3.2 discuss problems in this context for single machines and job shops, respectively.

2.2.3.1. Single Machine Problems

Applying simple dispatching rules like the EDD rule to the problem with nonsimultaneous arrival times may result in poor decisions due to its myopic nature. A branch and bound procedure can address explicitly the complex interactions between setup times and due dates. However the computational burden increases exponentially. Thus, Ovacik and Uzsoy [1994a] were motivated to seek intermediate methods, which lead to the idea of a Rolling Horizon Procedure (RHP), where the dynamic scheduling problem is decomposed into a series of smaller subproblems of the same type. In RHP, whenever a machine becomes available, a subproblem is solved using forecasts of future job arrivals that are predicted to occur over a certain time period called as a forecast window. A branch and bound algorithm was developed to solve the subproblems optimally but the restricted size of the subproblems in the RHP reduced the computational burden of this procedure.

n jobs are given and each job *j* with a known release time r_j, a processing time p_j and a due date d_j. Setup time of s_{ij} is incurred when job *j* is processed immediately after job *i*. The jobs are indexed in order of increasing release times. The algorithm is given as follow:

Algorithm RHP

Step0 Let $t = r_1$, $S(t) = \{\phi\}$
Step1 Determine the set $K(t)$
Step2 Optimally schedule the jobs in $K(t)$.
- Select the next l, $l = \min(\lambda, |K(t)|)$, jobs in the optimal schedule to the subproblem and place them in the schedule.

- Let these jobs form the set *L*.
- Set *t* to the completion time of the last job scheduled in *L*, and $S(t) = S(t) \cup L$.
- If all jobs have been scheduled, stop. Else go to step 1.

t : the current decision point
S(t) : the set of all jobs scheduled until that time
K(t) : the candidate jobs considered at the current decision point *t*

This algorithm was applied to scheduling test systems in the final test phase of semiconductor manufacturing. The RHP consistently yielded better schedules than many other commonly used rules such as dispatching rules (EDD) or dispatching rules combined with a local improvement procedure. The computational effort required by the RHP was heavily affected by the choice of the parameters: the maximum number of jobs considered at any decision point, κ; the maximum number of jobs scheduled at each decision point, λ; and the length of the forecast window, T.

2.2.3.2. Job Shop Problems

As in single machine problems, using a local dispatching rule in job shop problem avoids the difficulty of obtaining timely shop status information beyond the workcenter of current decision. However, using only local information is highly myopic in nature and, therefore, may perform badly in the long run. The approach of predicting future events was proposed because it might allow the use of characteristics of the arriving job, resulting in an improved schedule.

Gere [1966] developed the 'Look-ahead' (LAH) heuristic which allowed intentional insertion of idle time into the machine needed by an extremely urgent job, 'critical job'. Gere found that the use of this heuristic improved the performance of some dispatching rules.

One of few areas in which future event prediction has been used is that of scheduling batch processing machines. Glassey and Weng [1991] used a heuristic to decide when to start a batch to minimize average waiting time, assuming the availability of information on the next L arrivals. The performance of two rules of batch services, static local rule and adaptive rule, was compared. Glassey and Weng commented that global factory status should be used to predict the arrival of next few batches and that if these predictions had small error, the heuristic proposed would still perform better than the static local rule.

Based on these previous related works, Ovacik and Uzsoy [1994b] exploited shop floor status information to schedule complex job shops with sequence dependent setup times. Many companies have implemented sophisticated Shop Floor Information System (SFIS) which can track WIP and machine status in real time. This information makes it possible for a scheduling system to take the entire shop status into account when developing a schedule. A decomposition methodology proceeded by decomposing the job shop into a number of workcenters. These were scheduled in order of criticality until all workcenter had been scheduled and a feasible schedule achieved. The solutions to the subproblems was integrated into a solution to the job shop problem by using a network representation which modeled the interactions between the workcenters.

Deterministic simulation was used to predict the arrival of jobs at the current machine. These predictions were combined with an optimization procedure to select the next job to be processed. The algorithm Ovacik and Uzsoy used is a follows:

Algorithm LA (Whenever a machine, *m*, becomes available)

Step 1 Determine the set, *J*, of jobs to consider and when they become available for processing on *m* within time window T

```
Step 2   Select a job in J to be processed next.
         If the job is available, schedule it.
         If the job is not available yet,
            Find the available job with the EDD that can be finished by
            the time the selected job arrives and schedule it.
            If not such job exists, keep the machine idle until the
            arrival of the selected job.
```

Four variants of this algorithm and three dispatching rules were compared (Table 2.6). To evaluate the proposed algorithms, three settings were considered: a classical job shop with sequence-dependent setup times, a semiconductor testing facility and a reentrant flow shop.

Table 2.6. The rules compared in Ovacik and Uzsoy's study

Rule	Brief explanation
LAJ	Select the job in *J* with the earliest the due date
LAO	Select the job in *J* whose operation on machine *m* has the EDD
LAJ (β)	• Schedule the jobs in *J* using the J-EDD dispatching rule. • Pick the first β jobs in this schedule. • Sequence them optimally and schedule the first job next.
LAO (β)	• Schedule the jobs in *J* using the O-EDD dispatching rule. • Pick the first β jobs in this schedule. • Sequence them optimally and schedule the first job next.
J-EDD	When a workcenter becomes available, choose the available job with the EDD
O-EDD	Choose the available operation with the earliest operation due date
ATCS	Apparent Tardiness Cost with Setups found in Bhaskaran and Pinedo[1991].

Results showed that the proposed algorithms consistently outperform myopic dispatching rules at the expense of very modest increases in computation time. Algorithm LAO(β) was the best performer of the seven in terms of average solution quality and robustness. The decline in schedule quality due to use of myopic dispatching rules was much greater for shops with reentrant product flows and high competition for capacity at key resources than for job shops with random routing. However, Ovacik and Uzsoy noted that

these procedures used global information in a limited way, making local scheduling decisions at individual machines. The development of more sophisticated algorithms that would develop optimal schedules over the entire time horizon was suggested as future research.

2.2.4. Supply Chain Scheduling

In the 1980s, manufacturing companies focused on their internal functions and discovered new technologies and strategies such as just-in-time, lean manufacturing, and total quality management. They placed relatively little emphasis on other organizations within their supply chain network. Recently, they discovered that effective supply chain management is the next step in order to increase profit and market share due to three major developments in global markets and technologies [Simchi-Levi, D. et. al., 2000]:

- Ever-increasing customer demands in areas of product and service cost, quality, delivery, technology and cycle time brought about by global competition
- The emergence of and greater acceptance of higher-order cooperative inter-organizational relationships
- The information revolution

Supply Chain Management (SCM) is defined as "the integration and management of supply chain organizations and activities through cooperative organizational relationships, effective business processes, and high levels of information sharing to create high-performing value systems that provide member organizations a sustainable competitive advantage". A *value system* is defined as "a connected series of organizations, resources, and knowledge streams involved in the creation and delivery of value to end customers". The objective of value systems is to position organizations in the supply chain to achieve the

highest levels of customer satisfaction and value while effectively exploiting the competencies of all organizations in the supply chain [Handfield, R.B. and E.L. Nichols, Jr., 2002].

In 1997 American companies spent about 10% of the United States gross national product (GNP) on supply related activity. Unfortunately, this huge investment typically included many unnecessary cost components due to redundant stock, inefficient transportation strategies, and other wasteful practices in the supply chain [Simchi-Levi, D. et. al., 2000]. SCM optimizes logistical performance at the inter-organizational level, and this means integrated management of the movement of materials from the raw materials suppliers across the chain to the end customer in a timely and cost-effective manner that meets customers' service requirement [Handfield, R.B. and E.L. Nichols, Jr., 2002].

In this context, Taylor, D. and D. Brunt [2001] suggested a "build-to-order" approach. If companies could provide custom-built product to order, as opposed to make to forecast, it could solve the major deficiencies of the current system:

- Redundant stocks would not occur
- Product would be sold without discounts
- Customer service levels would rise.

The actual risk of build to order is short-term volatility (i.e. what happens if no orders come in the first week of the month, but all arrive in the second week?). The risk is that sales might be lost if the order lead time exceeds the customer's waiting tolerance. The flaw is to be seen in the manufacturers' abilities to manage demand. Manufacturers need to understand and manage their demand, not simply react to incoming orders, and increases marketing efforts.

To embrace this new philosophy, organizations must be able to compete on the basis

of time besides superior cost, quality, delivery, and technology considerations. As customers increasingly focus on time-based performance, supply chain member organizations must be able to meet this challenge. Again, the question is how we can reduce cycle time. Opportunities for cycle time reduction exist on both an intra-organizational and inter-organizational basis. Several authors have shown time to be a highly effective area to focus overall improvement efforts because only a small percentage of the total cycle time has anything to do with "real work". The rest of the time is typically devoted to a wide range of counterproductive, time-consuming activities and events. The opportunity for improvement appears to be even greater in an inter-organizational supply chain environment. Identifying, improving, and/or eliminating these time-consuming activities represent one of the major SCM opportunity areas [Handfield, R.B. and E.L. Nichols, Jr., 2002].

Given the complexity of supply chains, development of DSS to assist decision-makers in the design and operation of integrated supply chains is becoming increasingly common. Like many complex business systems, SCM problems are not so rigid and well defined that the flexibility, intuition, and wisdom that is a unique characteristic of humans is essential to manage the systems effectively. However, there are many aspects that can only be analyzed and understood effectively with the aid of a computer. These DSS do not make decisions. Instead, they assist and support the human decision maker in the decision-making process. These DSS help identify opportunities for improvements across the supply chain [Simchi-Levi, D. et. al., 2000]. Specific technologies that may be utilized for an effective supply chain management DSS include: SQL interface, expert system rules, scheduling algorithms, linear programming capabilities, blocked scheduling, multi-site/multi-stage scheduling, graphical user interface, user definable database, available-to-promise, and

demand management [Handfield, R.B. and E.L. Nichols, Jr., 2002].

2.3. Virtual Factory Scheduling Research

2.3.1. Overview of Virtual Factory

There are many tools available for scheduling job shop problems. The Virtual Factory, developed at NC State, is one such tool that has been found to provide near-optimal solutions to industrial-sized problems in seconds. The Virtual Factory is an iterative simulation-based procedure, which solves deterministic problems. The idea for this simulation-based job shop scheduling algorithm was first proposed by Lawrence and Morton [1986] and Vepsalainen and Morton [1988]. Hodgson et al. [1998, 2000] further developed it and named it the Virtual Factory.

Hodgson et al. [1998] showed that, under certain conditions, the Virtual Factory provides optimal, or near optimal, schedules for the *N*-job, *M*-machine, maximum lateness problem where processing times, due dates and job routings are known; all jobs are available for processing; and the objective is minimizing maximum lateness, L_{max}. The Virtual Factory consists both of a scheduling algorithm and a lower bound.

2.3.1.1. Scheduling Procedure

The procedure in the Virtual Factory to sequence a job shop without sequence dependent setup time consideration consists of repeatedly simulating the system to be scheduled while simultaneously updating the job sequence based on the results of the previous simulation. During the first iteration, jobs are sequenced on machines in order of increasing slack. Let d_i be the due date of job i and p_{ij} be the processing time of job i on

machine *j*. Then the slack of job *i* on machine *m* is calculated as

$$Slack_{i,m} = d_i - \sum_{j \in m+} p_{ij} \qquad (2.4)$$

where *m*+ is the set of all operations subsequent to machine *m* on job *i*'s routing. Slack represents the latest possible time that a job can finish on a machine and still satisfy its final due date.

In general, however, this may not provide good results, since slack does not take queuing time into account. Thus, in subsequent iterations of the simulation, the queuing time of the previous iteration is used to modify slack, and jobs are sequenced in order of revised slack. The revised slack for job *i* on machine *m* is computed as

$$Slack'_{i,m} = d_i - \sum_{j \in m+} p_{ij} - \sum_{j \in m++} q_{ij} \qquad (2.5)$$

where m++ is the set of all subsequent operations to machine *m* on the routing sheet for job *i,* except the immediate subsequent operation. Revised slack represents the time at which the job must finish processing in order to be available to start processing at its next operation. The simulation is then rerun using the revised slack from the previous iteration. The process is repeated until the estimates of the queuing times stabilize, or the lower bound is achieved, and the best solution is saved.

2.3.1.2. Lower Bound

One way of evaluating the quality of a solution is to compare it with a computed lower bound, *LB*. The lower bound is calculated by decomposing the job shop problem into individual one machine problems. To calculate the *LB*, the earliest possible start time ($ES_{i,m}$) for each job *i* on machine *m* and the latest possible finish time ($LF_{i,m}$ or slack) are computed.

$LF_{i,m}$ can be interpreted as the effective due date for job i on machine m and $ES_{i,m}$ as the release time for job i on machine m. Let r_i be the release time of job i. Then the earliest possible start time for a job i on machine m is

$$ES_{i,m} = r_i + \sum_{j \in m-} p_{ij} \qquad (2.6)$$

where $m-$ is the set of all operations preceding machine m on job i's routing sheet. The latest finish time for each job i on machine m is

$$LF_{i,m} = d_i - \sum_{j \in m+} p_{ij} \qquad (2.7)$$

where $m+$ is the set of all operations following machine m on the routing sheet of job i.

Solving the $N/1/ L_{max} \mid r_i$ problem on machine m provides a LB for the $N/M/ L_{max}$ problem. Since $N/1/ L_{max} \mid r_i$ is NP-hard, a relaxation suggested by Baker and Su [1974] is used. The relaxation is to allow preemption of a job in process whenever one with a more imminent due date becomes available.

The overall lower bound, LB (L_{max}), is computed as

$$LB(L_{\max}) = \max_{m=1,M} \{LB_m(L_{\max})\} \qquad (2.8)$$

where LB_m (L_{max}) is the lower bound for machine m. The power of this lower bound is that there are M chances to get a tight bound.

2.3.1.3. Experimental Results

The Virtual Factory was tested on randomly generated problems and two data sets from a large furniture manufacturing plant. For the randomly generated problem, a parameterized structure for generating job shop scheduling problems was used. Due dates were determined using two parameters: the expected number of tardy jobs (the tightness of

the due dates), T, and the due-date range parameter, R, as suggested by Demirkol et al. [1998]. If the range is held constant, the tightness of the due-dates does not affect the optimal sequence. The range of due dates has considerable effect on the performance of the Virtual Factory for a particular problem specified by the number of jobs, number of machines, number of operations per job, and processing time range. A pattern of the Virtual Factory's performance was shown in a graph with the difference between L_{max} and LB plotted against the due date range, R. The difference between L_{max} and LB represents the maximum by which the Virtual Factory solution could exceed the optimal solution. Hodgson et al. [1998] also found that the difference for the Virtual Factory was roughly equal to the expected queuing time of the L_{max} job. The procedure appeared to perform better on larger problems. The reason was conjectured that the LB calculation is stronger for larger problems.

2.3.1.4. Hot Job Acceleration

The research on idle time insertion by Hodgson et al. [2000] reported that a critical job (i.e., a job whose lateness is equal to or close to L_{max}) may be delayed by a non-critical job already in process when it arrives at a machine for processing. Thus, under certain conditions, improvements can be made in the schedule by inserting idle time into a machine schedule just prior to the arrival of a critical job at the machine. In other words, the non-critical job is held out of production, thus creating idle time on the machine, and allowing the critical job to start processing immediately upon arrival.

2.3.2. Sequence Dependent Scheduling

2.3.2.1. Scheduling Procedures

Taner [2001] modified the revised slack computations in the Virtual Factory to include sequence dependent setup considerations.

$$slack''_{i,m} = d_i - \sum_{j \in m_i^+} p_{i,j} - \sum_{j \in m_i^+} s^+_{i,j} - \sum_{j \in m_i^{++}} q_{i,j} \quad (2.9)$$

where s_{ij} is the setup time on machine j to process job i and s^+_{ij} is the part of this setup time that is incurred after the arrival of job i at machine j.

Following Ovacik and Uzsoy [1994], a forecasting mechanism, which was discussed in detail in section 2.2.3.2, was incorporated into Policy 2 which was the best scheduling rule based on the Taner [2001]'s study. Let τ be a lower limit on the length of time that a job should have until its operation due date (equivalent to *slack''* in the Virtual Factory) for it to be considered not urgent. Let γ be an upper limit on the length of time that a job should have until its operation due date for it to be considered extremely tardy. The algorithm is given as follows:

Policy 2 algorithm

Case 1: There is no job in the queue → Use SST
Case 2: There is only one job in the queue →

- If the job in the queue has τ time units or more until its due date and it has a bigger type index than the current setup, use SST
- Otherwise, use EDD

Case 3: There is more than one job in the queue →

- If all jobs waiting in the queue are already tardy and at least one is tardy by γ or more time units, use the SUBSET rule.
- Else if SST prefers a job in the queue which has τ time units or more until its due date while a more urgent job is also waiting for service, use EDD
- Else if SST and EDD prefer the same job in the queue for subsequent processing use EDD
- Use SST under all other conditions.

***SST: Shortest Setup Time**

- If there is at least one job of the type of the current setup in the queue, process the next job of that type.
- If the current setup is for the smallest index and there are jobs of larger index in the queue, process the next job which have the smallest index among the queue.
- If there are jobs of smaller and larger index in the queue than the current setup type, process the next job of the smallest index among larger index in the queue.
- If the current setup is for the largest index and there are jobs of smaller index in the queue, process the next job of the smallest index among the queue.
- If there are jobs of only one type in the queue, process the first in the queue.
- If there is nothing in the queue maintain the latest setup

***SUBSET**

- Form priority groups according to due dates
  - If the time until the due date for a job is smaller than or equal to a pre-specified value, that job is classified as of "high priority"
  - Otherwise, it is classified as of "low priority".
- Use the SST rule within the priority groups.

2.3.2.2. Lower Bound

The *LB* for this problem is equivalent to establishing a *LB* for the asymmetric traveling salesman problem with time windows. This is extremely difficult to compute. Therefore, the performance of the technique was evaluated in relation to that of the results of previously proposed algorithms. As a benchmark, the LAO(β) rule reported by Ovacik and Uzsoy [1994] was used.

2.3.2.3. Experimental Results

Taner compared the performance of four schemes. Scheme 1 was one version of Policy 2. To determine if a job should be considered not urgent or extremely tardy, this version uses the length of time from the point the machine becomes available until the

operation due date (*slack"*) of a candidate job. Scheme 2 was another version of Policy 2. This version is essentially the same for those candidate jobs which are already in the queue at the time the machine becomes available. However when a candidate job is estimated to arrive after the machine becomes available, the length of time from the estimated arrival time of the candidate job until its operation due date is used to determine if that job should be considered not urgent or extremely tardy. Scheme 3 was the original LAO(β) rule and the modified LAO(β) rule was used as Scheme 4.

These schemes were applied to two general job shop settings. The first one was a classical job shop where all machine have sequence dependent setups and each job visits every machine in some random order. In the second setting, only one of the machines requires sequence dependent setups, each job goes through this machine at the mid-point of its randomly generated operation route. In either setting, Scheme 1 and 2 performed significantly better and run only a fraction of a second slower than Scheme 3 and 4 on the average. The difference among the performance of the four scheduling methods became more significant when more jobs competed for the same key resources in the shop.

2.3.3. Dynamic Scheduling

The Virtual Factory has been tested primarily under transient circumstances. In industry, though, running a plant until it is empty is rare. Instead, plants usually contain many different orders, with new orders arriving as older ones are completed. Scheduling is often performed on some regular basis, i.e. everyday. The best schedule is implemented until the plant is rescheduled. To evaluate how well the Virtual Factory might perform in industry, it was tested in a rolling horizon setting [Thoney et al, 2002b].

2.3.3.1. Scheduling Procedure

The following definitions are required for this section:

t - Current time in days

c_j - Completion time of job j

N - Total number of jobs

N_s - Total number of jobs starting in factory on first day

M - Total number of machines

UL - Upper limit of uniform distribution for number of operations

JR - Number of jobs released each day

RO - Number of operations for jobs released

DL - Length of a day

T - Total horizon length in days

w - Number of days in warm-up period

WIP - Work in process (number of days)

i - Number of iterations

$\overline{Mops}$ - The average number of operations that a single machine can process in a day

$\overline{Ops}$ - The average of operations that each job, which starts in the factory, has

$\overline{P}$ - The expected job processing time.

The algorithm for the rolling horizon scheduling procedure is given as follows:

Rolling Horizon Algorithm

```
Step 1. Initialize t = 0
        1.1 If t = w + 1, compute LB
        1.2 Release jobs whose r_i = t
```

```
        1.3 Run the Virtual Factory N_i iterations
        1.4 Implement the first day of the best schedule
        1.5 t = t + 1
        1.6 Continue from 1.1 until t = T
Step 2. Run the remainder of the best schedule until all jobs are
        finished
Step 3. Initialize i =1
        3.1 If c_i > w, determine if job i is the L_max job
        3.2 i = i + 1
        3.3 Continue from 3.1 until i = N
```

Step 1 releases the new jobs into the system, runs the Virtual Factory, and implements the first day of the best schedule. This procedure is repeated each day until the total number of days is reached. In step 2, the best schedule is run until all jobs are finished. This ensures that the scheduling procedure did not sacrifice the remaining jobs in the factory to yield a good schedule. In Step 3, the lateness for each job completed after the warm-up period is compared to the current maximum lateness.

2.3.3.2. Lower Bound

The *LB* is computed in the same manner as for the original VF, except that the *LB* for the rolling horizon schedule is computed after the warm-up period. The *LB* calculation includes both jobs that are currently in the factory after the warm-up period, with their remaining operations and processing times, and also those jobs that are released later during the complete horizon of the simulation. Therefore, even though there are multiple runs of the VF engine for the rolling horizon scheduling procedure, there is only one *LB* calculation.

2.3.3.3. Calculation of *JR* and N_s

To balance input orders and output products, the number of jobs released each day and the total number of jobs starting in the factory on the first day needs to be calculated. A well-defined calculation procedure was developed by Thoney et al. (2002b). *JR* is calculated by following formulas:

$$JR \approx \frac{(M)(\overline{Mops})}{RO}. \qquad (2.10)$$

$$\text{where } \overline{Mops} = \frac{DL}{\overline{P}} \qquad (2.11)$$

N_s was computed by following formulas;

$$N_s \approx \frac{(JR)(RO)(WIP)}{\overline{Ops}} \qquad (2.12)$$

$$\text{where } \overline{Ops} = (UL+1)/2 \qquad (2.13)$$

2.3.3.4. Experimental Results

Due date range has been shown to be a factor influencing solution performance in evaluating algorithms scheduling transient job shops [Demirkol et al., 1998]. Due date range might also be a factor in scheduling jobs shops in rolling horizon scenarios. Therefore, a due date range, *DDR*, was defined so that each job, *i*, was randomly generated a discrete uniform due date between r_i and r_i + *DDR*, where $r_i=0$ for jobs initially in the factory. Experimentations were conducted on different scenarios. The effects of varying the total horizon length, increasing the warm-up period, varying the number of jobs released, varying the number of operations, and varying the number of jobs released and the number of operations, simultaneously, were tested. The Virtual Factory has been shown to perform well

in a rolling horizon setting under a variety of different conditions.

2.3.4. Supply Chain Scheduling

Thoney et al. [2002a] expanded Virtual Factory to include batch processors. Including transportation elements in the fundamental model enabled the detailed scheduling of entire multi-factory manufacturing supply chains.

2.3.4.1. Scheduling Procedure

The sequencing procedure was generalized to include batch processors within the production system. The manner in which queuing time was used in the Virtual Factory is inappropriate for batch operations because prioritizing the jobs by increasing revised slack does not account for the way they interact in batch processing. Two methods of incorporating batch processors into the procedure were discussed. The following notation is required.

Q_i - Queuing time for job i at the batch processor

JA_i - Time job i arrives at batch processor's queue

BF_i - Time the machine on which job i is processed finished its previous batch

BA_i - Time the next batch begins processing after the arrival of job i

BB_i - Time the previous batch begins processing before the arrival of job i

It is important to note that BB_i and BF_i may or may not refer to the time at which the same batch processor begins and ends processing. The queuing schemes were as follows.

Queuing Scheme 1. $Q_i = max\ \{JA_i - BF_i\ , 0\}$

Queuing Scheme 2. $Q_i = \begin{cases} JA_i - BB_i & \text{if } JA_i \le (BA_i - BB_i)/2 \\ 0 & \text{otherwise} \end{cases}$

Queuing Scheme 1 tried to force the machine to begin processing as soon as possible and Scheme 2 tried to place a job in the right batch. Since Scheme 2 tended to dominate Scheme 1, Scheme 2 was used for subsequent experiments.

Furthermore, it was assumed that a transportation vehicle does not leave until it is full or there are no more jobs left upstream in the system. This is consistent with what happens in industry when the batch processors are vehicles, as full loads minimize transportation costs.

2.3.4.2. Lower Bound

The lower bound used was an enhancement of the classic lower bound calculation for the *N*-job, *M*-machine job shop in the Virtual Factory. Since multiple vehicles may transport jobs from one factory to another, transportation is a problem of batch processors in parallel. This problem is known to be NP-hard. Therefore, the problem was relaxed for efficient computation.

First, the machine-batching problem was considered since this avoided the complications arising from the deadhead-return trip in the vehicle-batching problem. To find a lower bound for parallel machine-batching, a capacity relaxation and the use of preemption were proposed. Let p be the processing time of a batch machine, and let t be the one-way travel time of a vehicle. Solving the parallel machine-batching problem with $p = 2t$ adequately modeled the utilization of the vehicles but did not provide an accurate estimate of L_{max}. Since jobs actually finished halfway through p, the actual L_{max} was obtained by subtracting t from the L_{max} of the parallel machine-batching problem.

Then, the lower bound was improved in several ways. These enhancements were based on the characteristics of the solution to the original batching problem that had been lost

in the relaxation process leading to the lower bound (i.e., release times, due-dates, and capacity profiles).

2.3.4.3. Problem Generation

The situation of *Two Factories in Series* is illustrated in Figure 2.11.

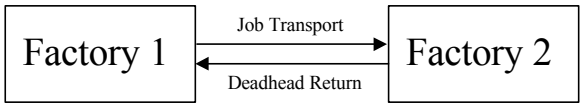


Figure 2.11. Scenario generated for *Two Factories in Series* [Thoney et al., 2002a]

Jobs were each assigned a number of operations, *n,* which was randomly generated, and a corresponding (Uniformly distributed) due-date. The number of operations was distributed Uniform [1, *UL*]. Recall that *UL* represents upper limit of uniform distribution for number of operations. If $n > (UL\text{-}1)/2 + 1$, then the job was processed on $n - \{(UL\text{-}1)/2 + 1\}$ machines in Factory 1, transported by truck to Factory 2, and processed on $(UL\text{-}1)/2$ machines in Factory 2. If $n = (UL\text{-}1)/2 + 1$, then the job was transported by truck to Factory 2, and processed on $(UL\text{-}1)/2$ machines in Factory 2. If $n < (UL\text{-}1)/2 + 1$, then the job was processed on n machines in Factory 2.

The situation of *Three Factories in Series* is illustrated in Figure 2.12.

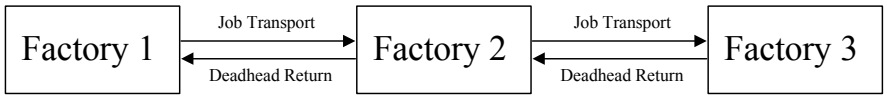


Figure 2.12. Scenario generated for *Three Factories in Series* [Thoney et al., 2002a]

Jobs were each assigned a number of operations, *n,* that was distributed Uniform [1, *UL*] and a corresponding due-date. If $n > 2\{(UL-2)/3\}+2$, then the job was processed on $n - [2\{(UL-2)/3\}+2]$ machines in Factory 1 and processed on $(UL-2)/3$ machines in Factory 2 and Factory 3. If $n = 2\{(UL-2)/3\}+2$, then the job was transported by truck to Factory 2, and processed on $(UL-2)/3$ machines in Factory 2 and Factory 3. If $(UL-2)/3+1 < n < 2\{(UL-2)/3\}+2$, then the job was processed on $n - \{(UL-2)/3+1\}$ machines in Factory 2 and processed on $(UL-2)/3$ machines in Factory 3. If $n = (UL-2)/3+1$, then the job was transported by truck to Factory 3, and processed on $(UL-2)/3$ machines in Factory 3. If $n < (UL-2)/3+1$, then the job was processed on n machines in Factory 3.

The scenario of *Two Factories Feeding One* is depicted in Figure 2.13.

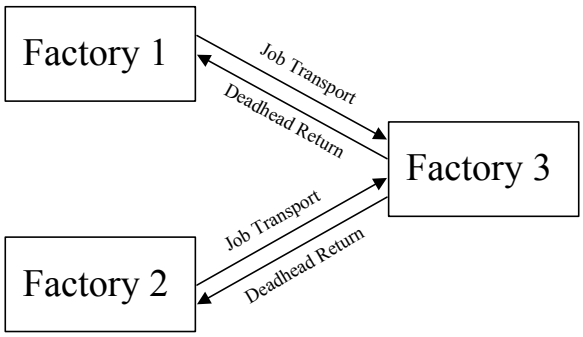


Figure 2.13. Scenario generated for *Two Factories Feeding One* [Thoney et al., 2002a]

A random number of operations, *n*, that was distributed Uniform[1, *UL*] was generated. If $n > (UL-1)/2 +1$, then 3 jobs were generated. The first job was processed on $n - \{(UL-1)/2 +1\}$ machines in Factory 2 and transported by truck to Factory 3. The second job was processed on $n - \{(UL-1)/2 +1\}$ machines in Factory 2 and transported by truck to Factory 3. The third job represented the assembly of 1 and 2 and is processed on $(UL-1)/2$ machines in Factory 3.

If $n = (UL\text{-}1)/2 + 1$, 3 jobs were also generated. The only difference was that the first and second job were not processed in Factory 1 and 2, respectively. If $n < (UL\text{-}1)/2 + 1$, then 1 job was generated. It was processed on n machines in Factory 3.

The scenario of *One Factory Feeding Two* is depicted in Figure 2.14.

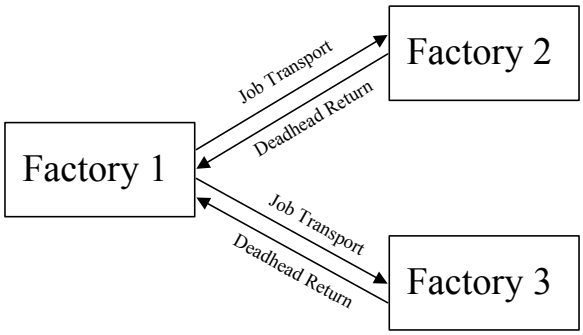


Figure 2.14. Scenario generated for *One Factory Feeding Two* [Thoney et al., 2002a]

A random number of operations, n, that was distributed Uniform[1, UL] was generated. If $n > (UL\text{-}1)/2 + 1$, then 3 jobs were generated. The first job was processed on $n - \{(UL\text{-}1)/2 + 1\}$ machines in Factory 1. It was split into the second and third job. The second job was transported by truck to Factory 2 and processed on $(UL\text{-}1)/2$ machines there. The third job was transported by truck to Factory 3 and processed on $(UL\text{-}1)/2$ machines there. If $n = (UL\text{-}1)/2 + 1$, 2 jobs were generated. The first job was transported by truck to Factory 2 and processed on $(UL\text{-}1)/2$ machines there. The second job was transported by truck to Factory 3 and processed on $(UL\text{-}1)/2$ machines there. If $n < (UL\text{-}1)/2 + 1$, then 2 jobs were generated. The first job and the second job were processed on n machines in Factory 2 and 3, respectively.

2.3.4.4. Experimental Results

The scheduling system performed well relative to the lower bound in scenarios in which transportation was not a bottleneck. In the *Two Factories in Series* scenario, the point at which trucks were no longer a bottleneck occurred when the average inter-factory transportation rate of the trucks was greater than or equal to the output rate of Factory 1.

Much of the bad performance was due to transient effects occurring at the beginning and the end of the schedules, which was not taken into account by the lower bound calculation. Using a long-run implementation of the system (i.e., rolling schedule) was conjectured to potentially eliminate much of the transient effect.

Thoney et al. [2002a] stated that the procedure was sufficiently efficient as to allow detailed scheduling of large-scale manufacturing/logistic supply chains in near-real time. The system provided the planner/scheduler with a 'look ahead' tool that could identify resource issues both in manufacturing and in transportation.

CHAPTER 3. ROLLING HORIZON SCHEDULING OF MULTI-FACTORY SCENARIOS

3.1. Introduction

Recently, opportunities for cycle time reduction which exist on an inter-organizational basis have received much attention. The potential for improvement appears to be even greater in an inter-organizational supply chain environment compared to an intra-organizational environment. Most of this analysis has been done at the macro level. Little research exists on the impact of coordinated, detailed production scheduling between different entities in the supply chain and the intermediate transportation.

Thoney et al. [2002a] expanded the VF to include inter-factory transportation operations which enabled the detailed scheduling of entire multi-factory manufacturing supply chains. Although performance was found to be good when transportation was not a bottleneck, the scenarios were tested in a transient setting. Starting and ending effects were observed to impact performance. The more realistic rolling horizon setting explained in Thoney et al. [2002b] would enable us to more accurately test how the VF would perform in multi-factory settings in industry by helping to eliminate transient effects. Using the rolling horizon algorithm, a variety of experiments were undertaken to gauge performance under different conditions.

In Section 3.2, 3.3, and 3.4, *One Factory*, *Two Factories in Series* and other multi-factory problems including *Three Factories in Series*, *Two Factories Feeding One* and *One Factory Feeding Two* are discussed, respectively. Section 3.5 summarizes the results of rolling horizon scheduling of multiple factory problems.

3.2. One Factory

3.2.1. Problem Generation of Base Case

Before testing the VF for multiple factory scenarios on a rolling horizon basis, the *One Factory* problem was generated and results were compared with previous research [Thoney et al., 2002b]. A 3 operation problem was generated since the number of factory operations for a *One Factory* problem would be doubled or tripled for multiple factory scenarios (i.e., 7 operations for *Two Factories in Series* and 11 operations for *Three Factories in Series* after adding inter-transportation operations). Parameters of *M* and *UL* varied in the problems (Table 3.1). For all base cases, *DL* = 1600 time units, *T* = 100 days, *w* = 10 days and *i* = 100. Recall that these parameters were defined in Section 2.3.3.1.

Table 3.1. The parameters for the base case

	M	*UL*
One Factory	25	3
Two Factories in Series	50	7
Three Factories in Series	75	11
Two Factories Feeding One	75	7
One Factory Feeding Two	75	7

In previous research [Thoney et al., 2002b], all jobs which were still in process at the end of the day were preempted and put back in queue, after adjusting the job's processing time to the remaining processing time. For the *One Factory* problem in this chapter, a different approach was taken. Those jobs which were still in process at the end of the day were not preempted and finished their remaining processing time at the beginning of the next day. Experiments were run for due date ranges between 0 and 25 days. For each due date range, 10 replications were run and the average difference between L_{max} and *LB* was calculated. This difference is the maximum by which the simulation solution could exceed

the optimal solution. A positive difference between L_{max} and LB could be the result of a non-optimal schedule, a weak LB, or a combination of the both.

3.2.2. Calculation of JR and N_s

For the rolling horizon scenario, the number of jobs released each day, JR, and the total number of jobs starting in the factory on the first day, N_s, were calculated based on previous research [Thoney et al., 2002b] as below (also refer to Section 2.3.3.3):

$$JR \approx \frac{(M)(\overline{Mops})}{RO}$$

$$N_s \approx \frac{(JR)(RO)(WIP)}{\overline{Ops}}$$

Since the processing times for the problems were uniformly distributed between 1 and 200, $\overline{P} \approx 100.5$. WIP was set equal to 3 days. Thus $JR \approx$ (25)(1600/100.5)/3=132.67 and $N_s \approx$ [(132.67)(3)(3)]/2 $\approx$ 600. Previous research showed that the calculated JR value overestimates the true number of jobs that can be released since no idle time on the machines was assumed. Therefore, experiments were performed to determine the maximum JR at which steady state could be achieved. Whenever the JR value was varied, correspondingly the N_s value was recomputed. Since the results with different due date ranges were similar, a due date range of 0 was used, and the number of jobs left in the system each day in the horizon was monitored. From visual inspection of the results, the day after which steady state is achieved was found. Then, the value at which the system stabilized was computed by averaging the numbers of jobs left in the system after that day. The maximum JR was found to be 128 jobs, and Figure 3.1 shows the results with a JR of 128 and a N_s value of 576 jobs.

The number of jobs left in the system increased until steady state was achieved after day 80 as shown in the graph.

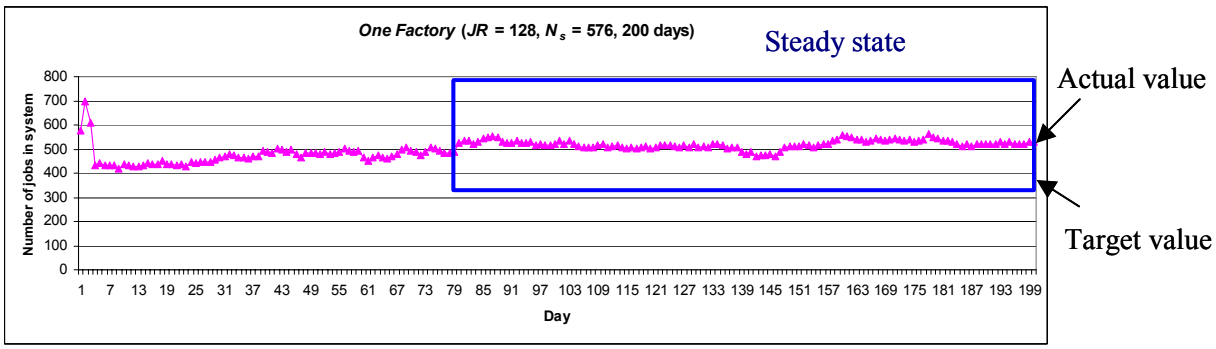


Figure 3.1. The number of jobs in the system with *JR* of 128 and N_s of 576

When conducting these experiments to determine the maximum *JR*, two concerns about the formula for the N_s computation were raised. The first concern was that the number of jobs in the system at steady state was not consistent with the results expected from using the N_s value. Since the N_s value was designed to achieve 3 days of *WIP*, the number of jobs in the system at steady state was expected to be around (*JR*×3) jobs. For example, with a *JR* of 128 and a N_s of 576 jobs, the target value is around 384 jobs but the actual steady state value was 520 jobs (Figure 3.1). This inconsistency between the number of jobs in which N_s is supposed to result at steady state and the actual value at which the system stabilizes was observed in most experiments.

The second concern about the N_s calculation was the amount of time needed to reach steady state. In many experiments using the N_s value, the steady state was not achieved until after many days had passed (i.e., 80 days in Figure 3.1). This would result in long runs with large warm-up periods to compute accurate statistics of steady state performance. Therefore,

finding the N_s value which stabilizes a system as fast as possible was advantageous. The VF was run with two starting N_s values; starting with almost no jobs and ($JR \times WIP$) jobs. Since the system seemed to become unstable when it was loaded near 100 % workload, a JR of 126 jobs was used to investigate how these N_s values affect the rate at which steady state is achieved. Figure 3.2 shows the number of jobs left in the system each day when JR is 126 jobs and N_s is 1 job, simulating that the factory was nearly empty on the first day. Figure 3.3 shows the number of jobs left in the system each day when JR is 126 jobs and N_s is 378 jobs. Steady state was achieved after 80 days and 52 days, respectively, as shown in the graphs. The values at which the systems stabilized were both 404 jobs. Even though the systems behaved differently, the values at which the systems stabilized were same.

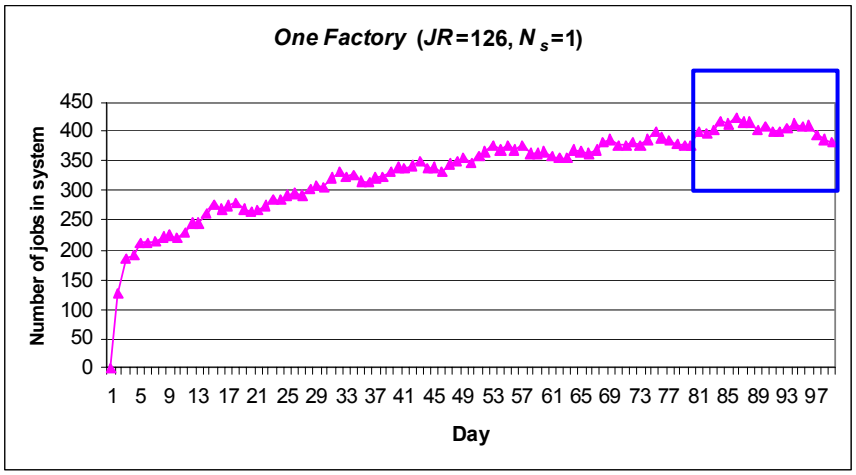


Figure 3.2. The number of jobs in the system with JR of 126 and N_s of 1

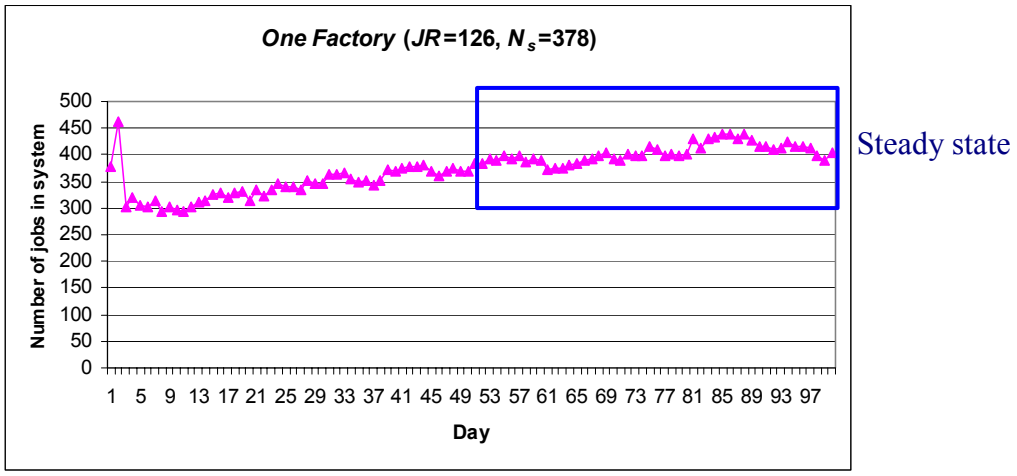


Figure 3.3. The number of jobs in the system with *JR* of 126 and N_s of 378

From these observations, the number of jobs starting in the system did not seem to be related to the value at which the system stabilizes, but it was related to how fast steady state is achieved. Since the N_s value computed by the formula did not work as anticipated and might not have achieved steady state as quickly as possible, we decided to find the value which achieved steady state fastest. Let the empirical N_s value be the number of jobs at which the system stabilizes. Intuitively, starting with the number of jobs at which the system stabilizes is the quickest way to reach steady state. For example, the empirical N_s value is approximately 520 jobs for a *JR* of 128 jobs (Figure 3.1) and 404 jobs for a *JR* of 126 jobs (Figure 3.2 and 3.3). Therefore, the empirical N_s value was used for further experiments instead of that computed by the N_s formula.

A regression curve of the number of jobs left in the system could be used to more easily determine the empirical N_s values for large numbers of different *JR* values than running experiments for each value of *JR*. In order to find the relationship between the *JR* value and the corresponding empirical N_s value, the VF was run with varying *JR* and a N_s

value of 1. This N_s value was chosen arbitrarily since the system will stabilize to the same value regardless of the initial value of N_s. Figure 3.4 displays the regression curve which is used for computing the empirical N_s values for various workloads.

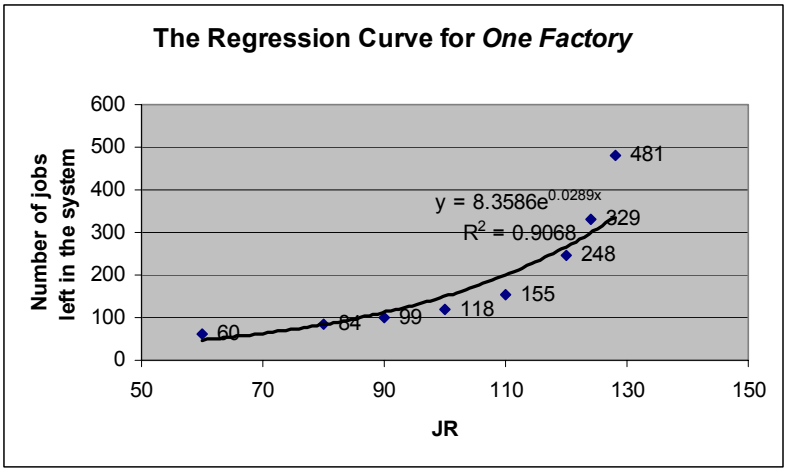


Figure 3.4. The regression curve for the empirical N_s value

Recall that the maximum value of *JR*, which satisfied steady state conditions, was found to be 128 jobs. However, since loading a plant to 100% capacity is rare in industry, 80%, 90%, and 95% workload were used in the experimentation. *JR* and the corresponding empirical N_s values used for further experiments are given in Table 3.2. Additional details about the calculation of the empirical N_s value can be found in Appendix A.

Table 3.2. *JR* and empirical N_s value for steady state in *One Factory*

Load %	*JR*	Empirical N_s
80	102	159
90	115	232
95	122	284

3.2.3. Performance Observations of Base Case

In Figure 3.5, the difference between the attained L_{max} and LB is displayed as a function of the due date range and workload. When the workload was 80%, the performance was similar regardless of the due date range. However, with 90% and 95% workload, the performance was somewhat worse in the low due date ranges and then became better in the middle and high due date ranges. This trend intensified as workload increased from 90% to 95%. To put these differences in perspective, recall that 90 days of factory performance were included in these statistics, with the latenesses of over (90)(102) = 9,180, (90)(115) = 10,350, and (90)(122) = 10,980 jobs taken into account for 80%, 90%, and 95% workload, respectively.

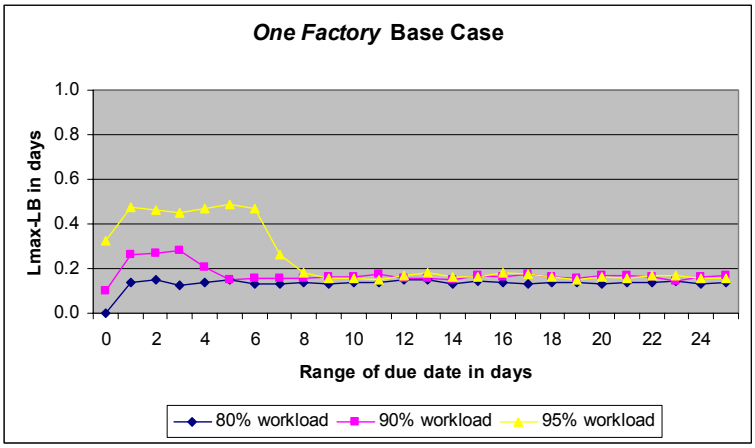


Figure 3.5. Performance of the base case of *One Factory*

3.2.4. Varying the Total Horizon Length

In order to determine the effect of the total number of days that are scheduled on the quality of the scheduling solutions, each problem was run for 55 days and 190 days with the same 10 day warm-up period. This allowed us to compare performances for half and twice as

long as the base case. The performance difference from the base case, ΔP, is defined as the difference between the ($L_{max} - LB$) computation for the base case and that of the case where one parameter is varied, for a specific workload value. This difference is shown as follows:

$$\Delta P = (L_{max} - LB)_{\text{Base case, workload}} - (L_{max} - LB)_{\text{Varying parameter, workload}} \quad (3.1)$$

Positive values mean that the performance improves compared to the performance for the base case and negative values mean that it worsens. This measure was designed to compare the effects of varying parameters more directly for the same workload, since it was adjusted according to the base performance.

For varying the total horizon length, ΔP was calculated by subtracting the ($L_{max} - LB$) computation for varying T from that of base case, T=100 days, for each workload value.

$$\Delta P_{\text{Varying } T\text{, workload}} = (L_{max} - LB)_{T=100\text{, workload}} - (L_{max} - LB)_{\text{Varying } T\text{, workload}} \quad (3.2)$$

For example, the performance difference from the base case for T=55 days and 80% workload was calculated as follows:

$$\Delta P_{T=55,\ 80\%} = (L_{max} - LB)_{T=100,\ 80\%} - (L_{max} - LB)_{T=55,\ 80\%}$$

In Figure 3.6, ΔP is displayed for varying T as a function of the due date range, total horizon length and workload. With 80% and 90% workload, ΔP was around zero, implying that performance of varying T was very similar to that of the base case regardless of total horizon length. Varying T had a more significant effect with 95% workload, especially in the low due date ranges. As T became larger, performance became somewhat worse.

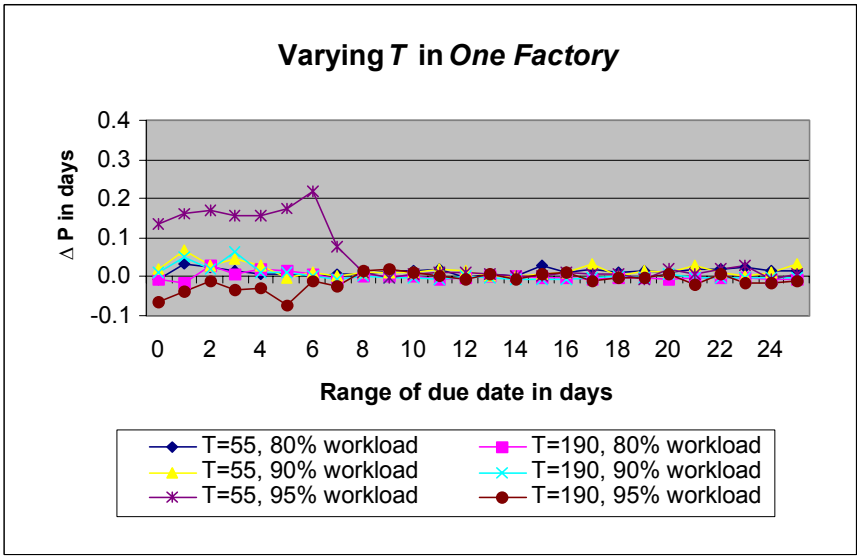


Figure 3.6. Performance with varying total horizon length in *One Factory*

3.2.5. Varying the Warm-up Period

The warm-up period was varied to be 20 and 40 days to determine how scheduling performance is affected. Each problem was run for 110 days and 130 days, respectively, in order to include 90 days of statistics. The performance difference from the base case was calculated by subtracting the ($L_{max} - LB$) computation for varying w from that of the base case, w =10 days, for each workload.

$$\Delta P_{\text{Varying } w, \text{ workload}} = (L_{max} - LB)_{w=10, \text{ workload}} - (L_{max} - LB)_{\text{Varying } w, \text{ workload}} \qquad (3.3)$$

In Figure 3.7, ΔP is displayed as a function of the due date range, warm-up period, and workload. The pattern of results was similar to that of varying the total horizon length. When the workload was 80% and 90%, ΔP values were around zero. Thus, there was no significant effect of varying the warm-up period. With 95% workload, increasing the warm-up period from 10 days to 20 or 40 days seemed to improve the scheduling performance in the low due

date ranges. Figure 3.8 and 3.9 show the 95% confidence intervals on ΔP when the warm-up period is 20 and 40 days, respectively, at 95% workload. From the graphs, the difference between the results when the warm-up period is 20 and 40 days is not statistically significant.

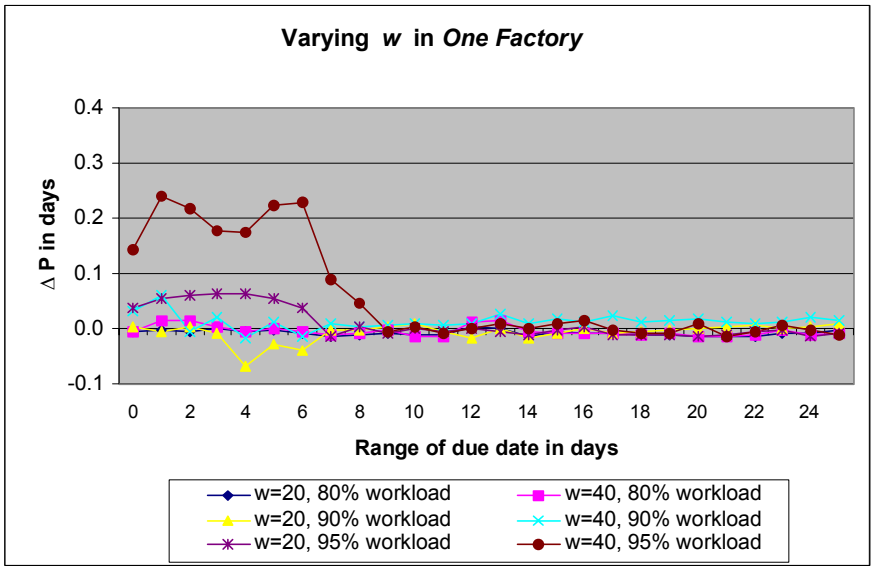


Figure 3.7. Performance with varying warm-up period in *One Factory*

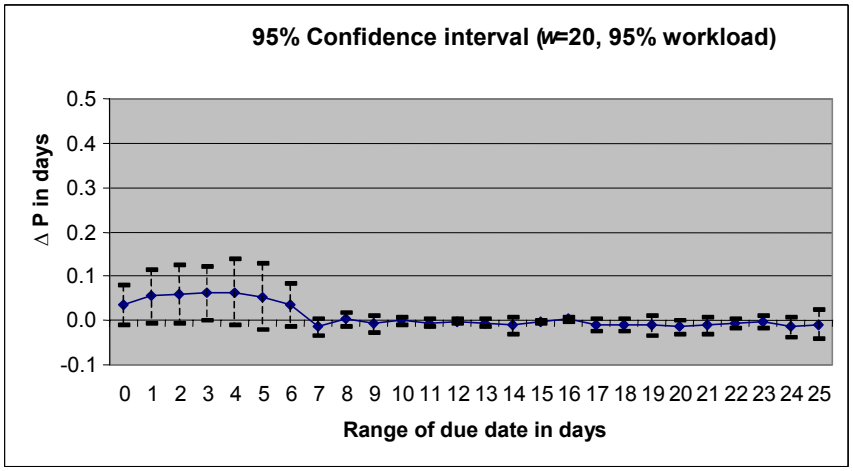


Figure 3.8. 95% confidence interval of $w = 20$ days and 95% workload in *One Factory*

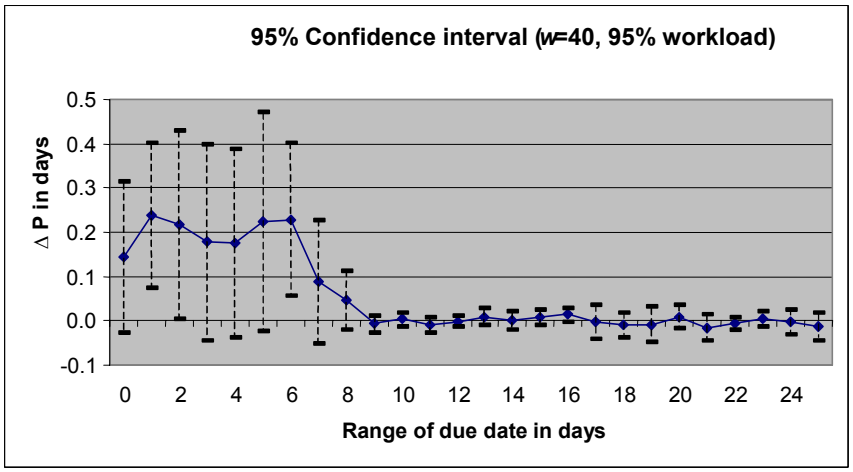


Figure 3.9. 95% confidence interval of w = 40 days and 95% workload in *One Factory*

3.3. Two Factories in Series

3.3.1. Problem Generation of Base Case

The parameters for the problem are given in Table 3.1. For job route generation details, refer to Section 2.3.4.3. Unlike the *One Factory* problem, the multi-factory problems include transportation operations between factories, thus corresponding parameters were added. One way travel time for trucks was 200 time units, and each truck had a capacity of 20 jobs. *JR* and an empirical N_s value for steady state in *Two Factories in Series* were found by the same procedure described in Section 3.2.2. Those values are given in Table 3.3. Since the empirical N_s value is for each factory, 2 N_s jobs started in the system initially. This problem implies that, for each problem generated, $2N_s/7$ jobs are expected to start with each of the possible number of operations (1-7). In addition, $2N_s(3/7)$ jobs are expected to start in each of the two factories.

Table 3.3. *JR* and empirical N_s value for steady state in *Two Factories in Series*

Load %	*JR*	Empirical N_s
80	102	270
90	115	396
95	122	486

3.3.2. Experimentation of Queuing Schemes

The queuing schemes for transportation were tested on the base case of *Two Factories in Series* to see how they performed in a rolling horizon setting. Recall that queuing scheme 1 tries to force the machine to begin processing as soon as possible and scheme 2 tries to place a job in the right batch (Refer to Section 2.3.4.1). The (L_{max} – LB) computation for queuing scheme 1, QS_1, is compared to that of queuing scheme 2, QS_2 as a function of the due date range and number of trucks (Figure 3.10). Unlike the results in the transient setting [Thoney et al., 2002a], there is no significant dominance between queuing scheme 1 and queuing scheme 2. In this chapter, scheme 2 will be used for all subsequent experiments. This follows Thoney's research which used scheme 2 for her subsequent experiments due to the dominance of scheme 2 over scheme 1.

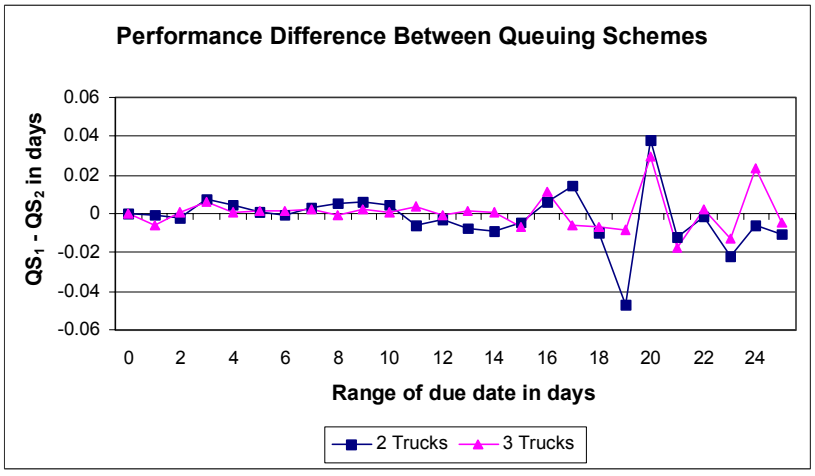


Figure 3.10. Performance comparison between queuing schemes

3.3.3. Performance Observations of Base Case

The calculations for estimating the minimum number of trucks for rolling horizon scenarios, so that the system will not blow up (i.e. input rate is not greater than output rate), is given in the following equation:

$$\left\lceil \frac{Round\ trip\ travel\ time\ of\ truck}{(DL/JR) \times Truck\ capacity} \right\rceil \tag{3.4}$$

This formula is closely related to bottleneck calculations used for transient problems [Thoney et al., 2002a], in which performance was very similar no matter how many trucks were added when trucks were not a bottleneck. Since the factory input is balanced with the factory output, when 128 jobs are released each day, a job is finished in Factory 1 on average every 1600/128 ≅ 12.5 units of time. Therefore, on average, a new truckload (truck capacity = 20 jobs) is ready for loading in Factory 1 every 12.5(20)=250 time units. To handle this rate, when the round trip travel time for each truck is 400 time units, approximately $\lceil 400 / 250 \rceil = 2$ trucks are needed. Thus, the number of trucks was set to 2 and 3 for subsequent experiments to avoid unrealistic cases where jobs would build up due to a transportation bottleneck, causing the system to explode.

In Figure 3.11, the difference between the attained L_{max} and LB is displayed as a function of the due date range, workload and number of trucks. When the workload was 80%, the performance was similar regardless of the due date range. However, with 90% and 95% workload, the performance was somewhat worse in the low due date ranges and then became better in the middle and high due date ranges. This trend intensified as workload increased from 90% to 95%. Since there is little difference in performance between 2 and 3 trucks, the results with 2 trucks will be presented for subsequent experiments.

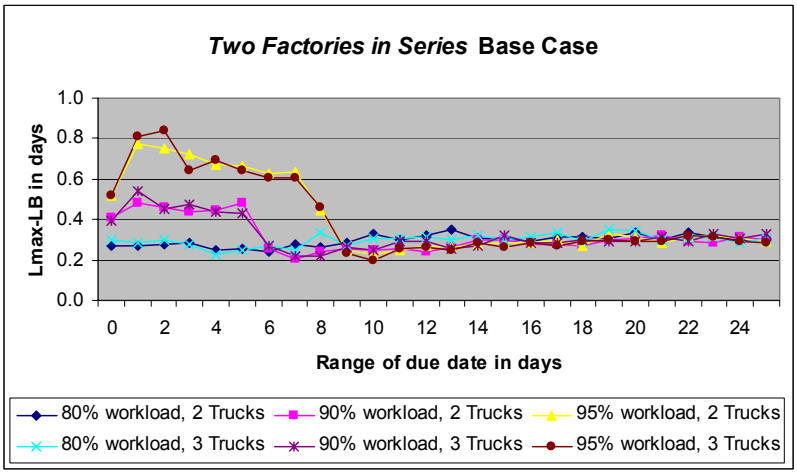


Figure 3.11. Performance of the base case of *Two Factories in Series*

The actual values of the *LB* and L_{max} are displayed in Figure 3.12. As due date range increased, both the L_{max} and *LB* values decreased linearly until the values reached a minimum. The point of change over the due date range for each workload in Figure 3.12 corresponds to the point after which L_{max} value was close to *LB* in Figure 3.11.

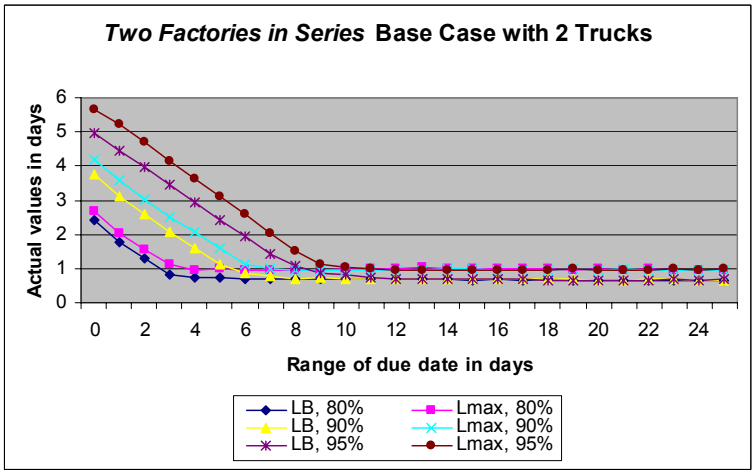


Figure 3.12. Actual *LB* and L_{max} values for base case of *Two Factories in Series*

Recall that a positive difference between L_{max} and LB could be interpreted as the result of a non-optimal schedule or a weak lower bound. One possible explanation for the positive difference in the low due date ranges could be related to the number of critical jobs whose lateness is equal to or close to L_{max}. A small due date range means that jobs released on a given day have similar due dates. This tends to create more critical jobs compared to when the due date range is large. The number of critical jobs may also increase as the workload increases, due to the increase in the competition among jobs for machines. This increase in the number of critical jobs could make it more difficult to schedule to minimize L_{max} and also could make the lower bound calculation weaker because more preemption may occur.

Each data point on the graphs represents the average of 10 problems and each problem was run for 100 iterations. Each of the problems in Figure 3.11 took an average of 218.62 seconds to run on a 2 GHz Pentium. Of this time, approximately 56.58 seconds was used in computing the lower bound, and it took about 1.62 seconds to schedule and implement the best sequence for each of the 100 days in the horizon. The computational effort required for the remaining experimentation in this dissertation is analogous, with the computation time per day increasing approximately linearly with the problem size.

3.3.4. Varying the Truck Capacity

Truck capacity was varied to be 1 and 10 jobs to determine how scheduling performance is affected. The performance difference from the base case was calculated by subtracting the ($L_{max} - LB$) computation for varying truck capacity from that of base case, *truck capacity* = 20 jobs, for each workload.

$$\Delta P_{\text{Varying } capacity, \text{ workload}} = (L_{max} - LB)_{capacity=20, \text{ workload}} - (L_{max} - LB)_{\text{Varying } capacity, \text{ workload}} \quad (3.5)$$

In Figure 3.13, ΔP with 40 trucks of capacity 1 and 4 trucks of capacity 10 is displayed as a function of the due date range when the factory is at 80% load. More trucks with smaller capacity showed somewhat better performance. Forty trucks of capacity 1 and 4 trucks of capacity 10 performed better than two trucks of capacity 20 by an average of 0.06 and 0.04 days, respectively. One explanation for the difference in performance is that there is extra waiting time when trucks have higher capacities, since it is assumed that a truck does not leave until it is full. This extra waiting time is not taken into account in the *LB*. When the factory workload is 90% and 95%, the pattern of results was similar to that of 80% workload. However, when truck capacity was set and workload value was varied, there was not a significant trend in performance.

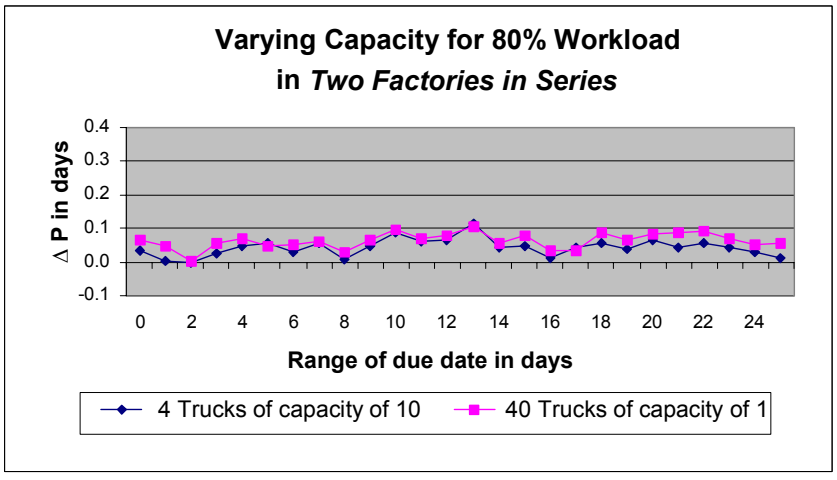


Figure 3.13. Performance with varying truck capacity at 80% workload in *Two Factories in Series*

3.3.5. Varying the Total Horizon Length

Each problem was run for 55 days and 190 days with the same 10 day warm-up period. ΔP was calculated by subtracting the ($L_{max} - LB$) computation for varying T from that of base case, T=100 days, for each workload value (Equation 3.2). In Figure 3.14, ΔP is displayed for varying T as a function of the due date range, total horizon length and workload. Regardless of workload value, smaller total horizon lengths performed better than the larger ones in the low due date ranges. This pattern is more evident in the *Two Factories in Series* problem compared to the *One Factory* problem (Figure 3.6).

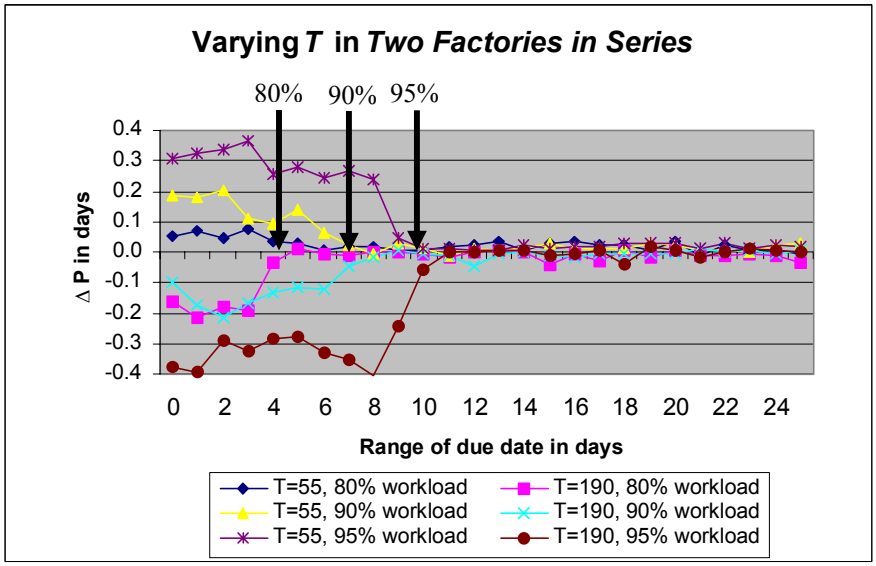


Figure 3.14. Performance with varying total horizon length in *Two Factories in Series*

Arrows in Figure 3.14 point to the location where ΔP becomes close to zero for each workload. This point occurred at a larger due date range as workload increased. For example, the performance of varying T became close to that of the base case when the due date range was greater than 5, 8 and 11 days at 80%, 90% and 95% workload, respectively. The absolute

values of ΔP also increased as workload increased with the total horizon length of 55 days and 190 days, respectively. It also can be interpreted that the effect of varying the total horizon length increased as the workload increased.

3.3.6. Varying the Warm-up Period

The warm-up period was varied to be 20 and 40 days. Each problem was run for 110 days and 130 days, respectively, in order to include performance of 90 days into the statistics, as was included in the original base case. ΔP was calculated by subtracting the ($L_{max} - LB$) computation for varying w from that of base case, w =10 days, for each workload (Equation 3.3). In Figure 3.15, ΔP is displayed as a function of the due date range and warm-up period when the factory is at 90% load.

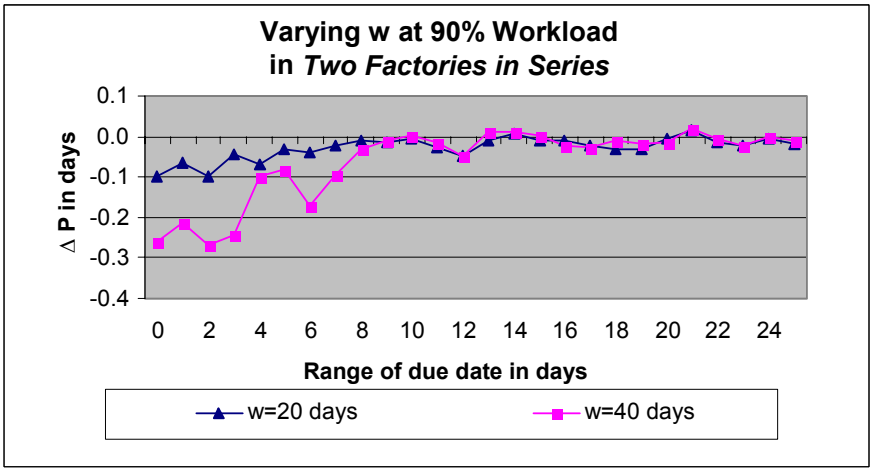


Figure 3.15. Performance with varying warm-up period at 90% workload in *Two Factories in Series*

When the workload was 80% and 90%, increasing the warm-up period from 10 days to 20 or 40 days resulted in a slight deterioration in the scheduling performance in the low due date ranges. Then, ΔP became close to zero in the middle and high due date ranges, similar to

the case of varying the total horizon length. However, at 95% workload, there was no significant pattern. This pattern is different, compared to the results of the *One Factory* problem (Figure 3.7).

3.4. Other Multi-factory Problems

The performance of *Three Factories in Series*, *Two Factories Feeding One* and *One Factory Feeding Two,* while not completely identical to *Two Factories in Series,* was similar. Thus results are not presented except for the base case of each scenario. For details of job route generation, refer to Section 2.3.4.3. Recall that the number of trucks displayed on each graph refers to the number of trucks that are transporting jobs between each location.

3.4.1. Three Factories in Series

In this problem, the parameters are the same as in the base case of *Two Factories in Series*, with the exception that the maximum number of operations is 11 (Table 3.1). *JR* and an empirical N_s value for steady state in *Three Factories in Series* were found by the procedure in Section 3.2.2 and are shown in Table 3.4. In this problem, 3 N_s /11 jobs are generated with each of the possible number of operations (1-11).

Table 3.4. *JR* and empirical N_s value for steady state in *Three Factories in Series*

Load %	*JR*	Empirical N_s
80	101	349
90	113	497
95	120	611

In Figure 3.16, the difference between the attained L_{max} and LB is displayed as a function of the due date range, workload, and the number of trucks. At 80% workload, the performance of *Two Factories in Series* was slightly better than that of *Three Factories in Series* in the low due date ranges. At 90% and 95% workload, the VF performed very similar in *Three Factories in Series* and *Two Factories in Series* (Figure 3.11).

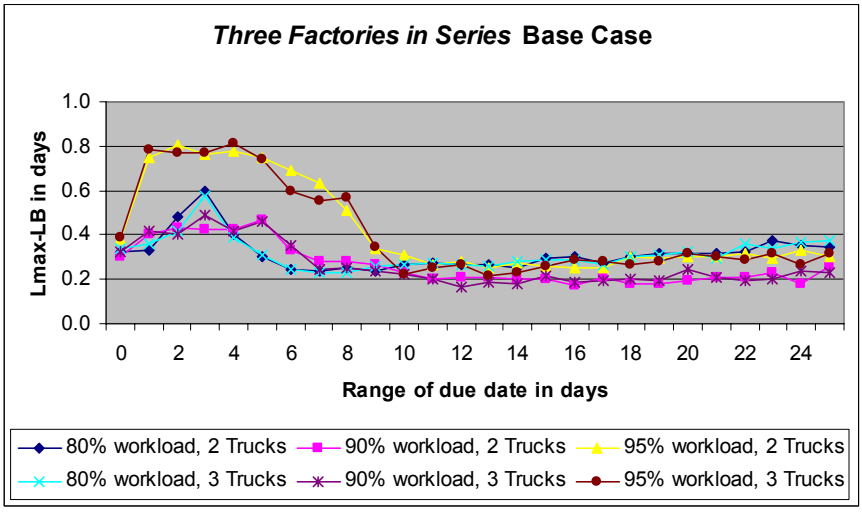


Figure 3.16. Performance of the base case of *Three Factories in Series*

3.4.2 Two Factories Feeding One

All parameters are the same as the base case of *Two Factories in Series* (Table 3.1). *JR* and an empirical N_s value for steady state in *Two Factories Feeding One* are given in Table 3.5.

Table 3.5. *JR* and empirical N_s value for steady state in *Two Factories Feeding One*

Load %	*JR*	Empirical N_s
80	96	232
90	108	313
95	114	364

In Figure 3.17, the difference between the attained L_{max} and LB is displayed as a function of the due date range, workload, and the number of trucks. Performance was somewhat worse in the low due date ranges but close to the lower bound after 7 days. Compared to the performance in *Two Factories in Series*, the difference between L_{max} and LB was slightly higher at 80% and 90% workload in the low due date ranges. Recall that in the problem generation, it was assumed that a certain job from Factory 1 was matched up with a job in Factory 2. Once both of these jobs arrived at Factory 3, they were replaced by a single job assembly. This potential added waiting time was not taken into account in the lower bound.

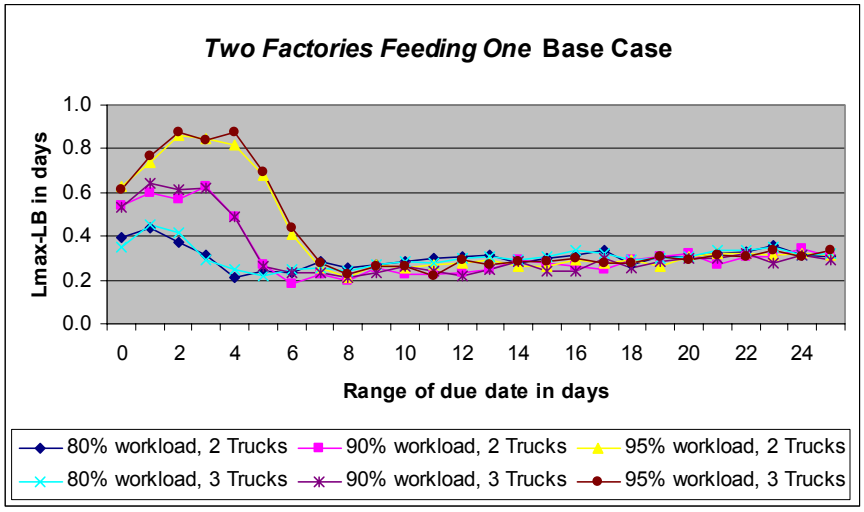


Figure 3.17. Performance of the base case of *Two Factories Feeding One*

3.4.3. One Factory Feeding Two

All parameters are the same as the base case of *Two Factories in Series* (Table 3.1). *JR* and an empirical N_s value for steady state in *One Factory Feeding Two* are shown in Table 3.6.

Table 3.6. *JR* and empirical N_s value for steady state in *One Factory Feeding Two*

Load %	*JR*	Empirical N_s
80	96	267
90	108	350
95	114	401

In Figure 3.18, the difference between the attained L_{max} and *LB* is displayed as a function of the due date range, workload and the number of trucks. Performance was somewhat worse in the low due date ranges and close to the lower bound after 5 days. Compared to the performance in *Two Factories in Series*, the difference between L_{max} and *LB* was slightly higher at 80% workload, similar at 90% workload and slightly lower at 95% workload in the low due date ranges.

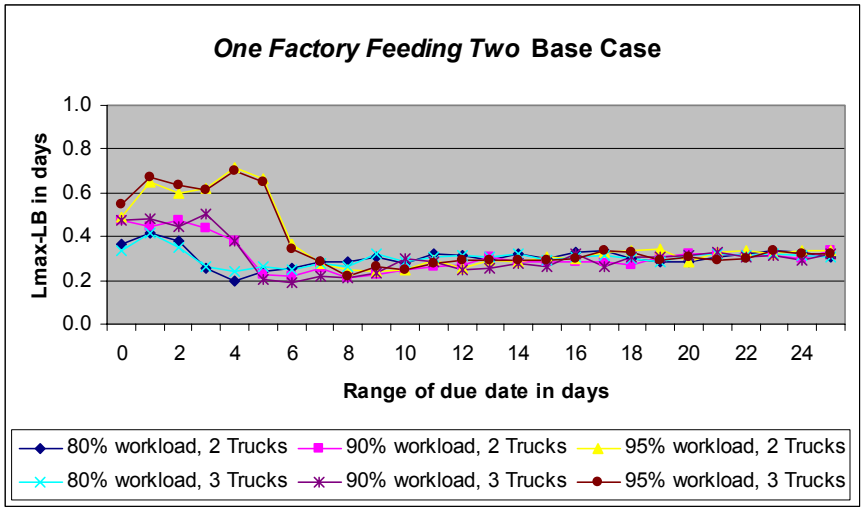


Figure 3.18. Performance of the base case of *One Factory Feeding Two*

3.5. Summary and Discussion

The performance of the VF has been tested in multi-factory, rolling horizon settings including *Two Factories in Series*, *Three Factories in Series*, *Two Factories Feeding One*

and *One Factory Feeding Two*. Considering the total horizon length of 100 days and the number of jobs processed, the VF performed well in all multi-factory supply chain environments in that the difference between L_{max} and *LB* was less than 1 day (Table 3.7). Compared to the *One Factory* problem, multi factory problems had a somewhat greater difference. Added waiting time for transportation between factories could be a reason. Moreover, *Two Factories Feeding One* could have extra waiting time for assembly.

Table 3.7. The maximum value of the difference in L_{max} - *LB* in days with 2 trucks

	80%	90%	95%
One Factory	0.15	0.28	0.49
Two Factories in Series	0.35	0.48	0.78
Three Factories in Series	0.59	0.46	0.81
Two Factories Feeding One	0.44	0.63	0.86
One Factory Feeding Two	0.42	0.47	0.71

Four simulation parameters, *JR*, truck capacity, total horizon length, and warm-up period, were varied in each problem. *JR* values were varied to be 80%, 90%, and 95% workload. Truck capacity was varied to be 1, 10 and 20 jobs. The effect of the total number of days was determined by varying it to be 55, 100 and 190 days. Also the warm-up period was varied to be 10, 20 and 40 days. Table 3.8 summarizes the relationship between the scheduling performance and the simulation parameters.

Table 3.8. The relationship between simulation parameters and the difference in L_{max} - *LB*

	Decrease leads to	Increase leads to
Due Date Range	Larger difference	Smaller difference
% Workload	Smaller difference	Larger difference
Truck capacity	Smaller difference	Larger difference
Total Horizon Length	Smaller difference	Larger difference

The effects of varying the parameters were tested, and there was some difference compared to the base case, especially in the low due date ranges. This difference was very small considering the total horizon length of 100 days and number of jobs processed in that the maximum values of Δ P was less than 0.4 days in all cases. Thus, the performance of the VF appears robust.

CHAPTER 4. CHARACTERIZING SETUPS

IN BATCHWISE FABRIC DYEING OPERATIONS

4.1. Introduction

According to Koksal et al. [1990], the dyeing operation could be decomposed into three subactivities, namely setup, operating, and unloading (Figure 4.1).

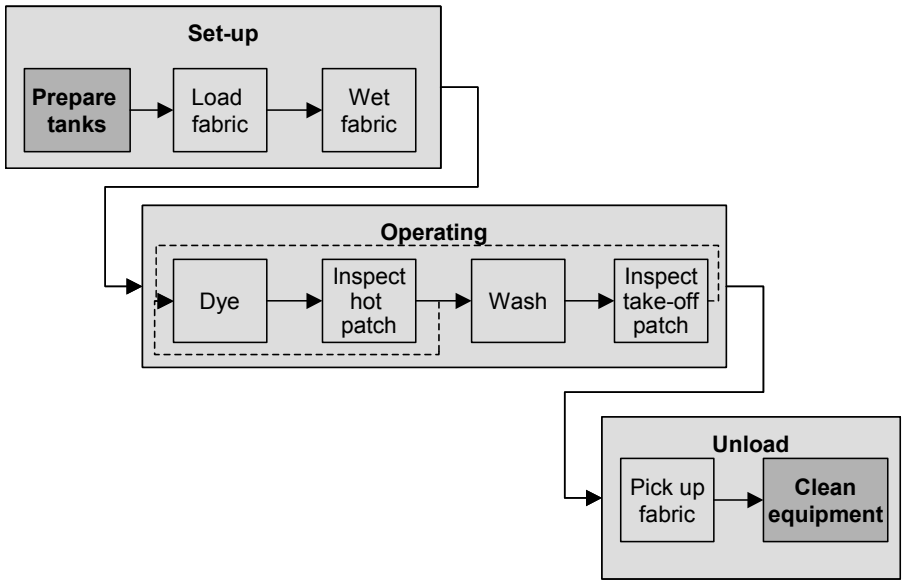


Figure 4.1. Functional decomposition of batchwise dyeing operation

Except for preparing the tank, all the subsequent processes occur on the same dyeing machine. Since the operations of loading the fabric into the machine and removing it each take several minutes or longer, it is common practice to perform a sequence of processes on the same machine, simply by replacing one treatment liquor with the next. Changeover from one color to another involves setting up the dye bath for a specific color. After unloading the fabric, the machine is boiled out of any previous dyestuff, if necessary. Dyeing a pale color

usually does not leave dyestuff residues because of its small usage of the dyestuff thus, the boiling up process is usually not necessary. However, the boiling up process is often needed after dyeing dark colors due to their dyestuff residues.

There are two ways to reduce setup time. One is to schedule the process of preparing the dye bath tank before the corresponding fabric's arrival or before the corresponding machine becomes available. Treating this preparing time separately from the processing time allows operations to be performed simultaneously and hence improves performance. The other is to sequence the jobs in an order which minimizes the number of equipment cleaning processes. Since loading and unloading are sequence independent and only equipment cleaning time can be saved by scheduling, only setups representing equipment cleaning processes will be considered.

The simplified setup matrix that Taner [2001] used for sequence dependent family setup times (Section 2.3.2) provides a basis for characterizing setups in batchwise fabric operations when families are indexed from lighter to darker colors. An interview with a representative of SK Chemicals, a leading textile company in South Korea, revealed that the assumptions that the setup times occurring in dyeing a lighter color after a darker color were fairly constant and independent of the exact colors were realistic. The biggest concern about Taner's matrix is that there are additional complications in applying it to dyeing operations. First of all, formulating color families is not an easy job in that a dyeing plant deals with a tremendous number of different colors, and it is not always obvious to which family a specific color belongs. Moreover, indexing a family is also not a simple task since it is hard to directly compare two different colors (i.e. red and blue) and tell which one is lighter or darker. Even worse, dyeing processes and the corresponding dyes vary with fabric type.

Since it is not common in industry to record setup time data for dyeing operations, it is difficult to collect related information and construct an empirical setup time matrix. Thus, a procedure of expanding Taner's setup time matrix was developed by taking the theoretical approach of color physics.

4.2. Family Formulating and Indexing

According to McDonald [1987], there are two ways of formulating color families. One is visual sorting. It is known as slow, expensive and the most difficult of all the visual assessments of color. The other is instrumental sorting. In 1976, Hunter Lab introduced a method of shade sorting based on the CIELAB cylindrical coordinates L^*, C^* and h recommended by CIE in May 1976. Known as hue angle sorting, this is effective because the volumes are oriented along the three variables of perceived color, i.e. lightness, chroma and hue. According to this approach, color coordinates of ordered jobs, given by the spectrophotometers, can be plotted in the cylindrical L^* C^* h space. Figure 4.2 shows the top view of the color space given by the chroma and hue angle.

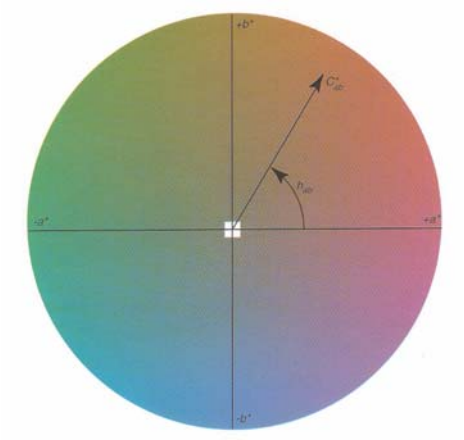


Figure 4.2. The top view of the color space given by chroma and hue angle

For family formulating, jobs were sorted by their hue angles and categorized in a counter-clockwise direction from Yellow. In this way, the color space in Figure 4.2 was divided into 8 hue indices or family indices, H, which are described in Table 4.1. The minimum and maximum value of each index will be denoted by *min value* (*Index*) and *max value* (*Index*), respectively.

Table 4.1. Eight hue indices in $L^* C^* h$ space

Hue index (Family index) H	Name	Hue angle $min\ value\ (H) \le h \le max\ value\ (H)$	
		min value (*H*)	*max value* (*H*)
1	Yellow (Y)	67.5	$112.4\overline{9}$
2	Yellow-green (GY)	112.5	$157.4\overline{9}$
3	Green (G)	157.5	$202.4\overline{9}$
4	Blue-green (BG)	202.5	$247.4\overline{9}$
5	Blue (B)	247.5	$292.4\overline{9}$
6	Violet (RB)	292.5	$337.4\overline{9}$
7	Red (R)	337.5	$22.4\overline{9}$
8	Orange (YR)	22.5	$67.4\overline{9}$

Then, a lightness index, L, and a chroma index, C, were given based on its lightness and chroma value, respectively. Lightness and chroma indices are shown in Table 4.2 and Table 4.3, respectively. In defining indices for lightness and chroma, Taner's convention of setting indices to develop lower triangular matrices will be used. A lower lightness index is defined to have lighter color and a higher lightness index is defined to have darker color. For chroma, since dull color requires mixed dyestuffs in the dye bath, it is usual to dye brighter colors first and then dull colors. Thus, a lower chroma index is defined to have stronger and brighter color and a higher index is defined to have duller color. Figure 4.3 shows the three indices of

a given job in cylindrical color space. Jobs closer to the bottom and core of the cylinder have a large lightness and chroma index, respectively.

Table 4.2. Ten lightness indices in $L^* C^* h$ space

Lightness index L	Lightness value $min\ value\ (L) \le L^* \le max\ value\ (L)$	
	min value (L)	*max value* (L)
1	90	100
2	80	$89.\overline{9}$
3	70	$79.\overline{9}$
4	60	$69.\overline{9}$
5	50	$59.\overline{9}$
6	40	$49.\overline{9}$
7	30	$39.\overline{9}$
8	20	$29.\overline{9}$
9	10	$19.\overline{9}$
10	0	$9.\overline{9}$

Table 4.3. Five chroma indices in $L^* C^* h$ space

Chroma index C	Chroma value $min\ value\ (C) \le C^* \le max\ value\ (C)$	
	min value (C)	*max value* (C)
1	40	50
2	30	$39.\overline{9}$
3	20	$29.\overline{9}$
4	10	$19.\overline{9}$
5	0	$9.\overline{9}$

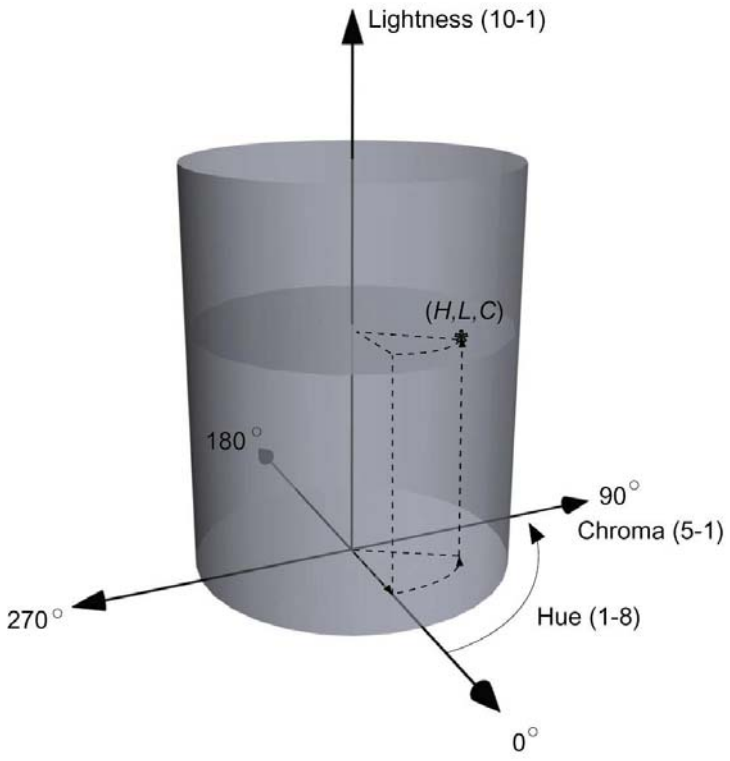


Figure 4.3. Three indices of a job in the color space

4.3. Setup Matrix Structure by Index

4.3.1. Setup Matrix by Fabric/dye Type Index

The setup structure varies based on four indices; fabric/dye type, hue, lightness and chroma denoted by $\{F, H, L, C\}$, respectively. Let s_F be the boil out time for a combination of fabric and dye group F. Figure 4.4 shows the setup time matrix for the difference between five fabric/dye indices. This matrix is independent of the hue, lightness, and chroma of the color being dyed. Since boiling out time depends on the dyeing profile of a combination of fabric and dye type on the machine, setup time is dependent upon the current fabric/dye type and is independent of the following fabric/dye type.

F\T	1	2	3	4	5
1	0	s_{F1}	s_{F1}	s_{F1}	s_{F1}
2	s_{F2}	0	s_{F2}	s_{F2}	s_{F2}
3	s_{F3}	s_{F3}	0	s_{F3}	s_{F3}
4	s_{F4}	s_{F4}	s_{F4}	0	s_{F4}
5	s_{F5}	s_{F5}	s_{F5}	s_{F5}	0

Figure 4.4. Fabric/dye type setup matrix

4.3.2. Setup Matrix by Hue Index

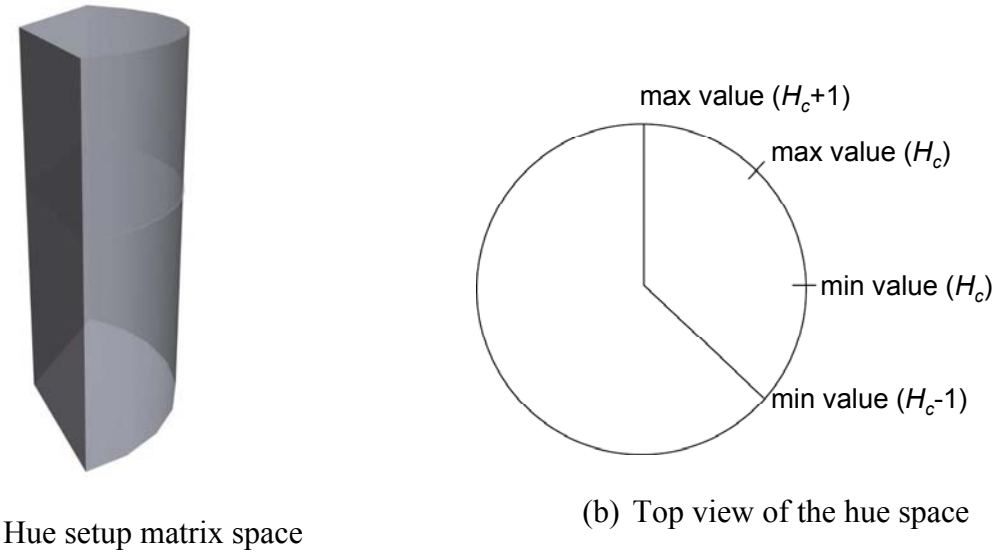
While the setup times between different fabric/dye type indices are independent of the colors dyed, the setup times for changes in hue, lightness and chroma are interdependent. In order to analyze the interdependency of the hue, lightness and chroma of colors dyed within the same fabric/dye type for determining setups, the problem is first approached graphically in three dimensional color space. Initially, each index will be analyzed separately. The intersection of the spaces formed by the treating the hue, lightness and chroma indices separately creates part of the space. Then, the interdependency will be treated. Given a color coordinate of a job, the limit at which the next color can be located without setup can be found in the color space.

Let the current job on the machine have the four indices $\{F_c, H_c, L_c, C_c\}$. Figure 4.5 shows the setup time matrix when setup is needed due to the difference between two hue indices when dyeing the same fabric/dye type. Setup is inserted unless the two indices are the same or adjacent. This is based on the human eye's tolerance.

F\ T	1	2	3	4	5	6	7	8
1	0	0	s_F	s_F	s_F	s_F	s_F	0
2	0	0	0	s_F	s_F	s_F	s_F	s_F
3	s_F	0	0	0	s_F	s_F	s_F	s_F
4	s_F	s_F	0	0	0	s_F	s_F	s_F
5	s_F	s_F	s_F	0	0	0	s_F	s_F
6	s_F	s_F	s_F	s_F	0	0	0	s_F
7	s_F	s_F	s_F	s_F	s_F	0	0	0
8	0	s_F	s_F	s_F	s_F	s_F	0	0

Figure 4.5. Hue setup matrix

Let hue space be defined as the color space which covers certain hue indices, all the lightness indices and all the chroma indices. It will be denoted by D_H [*starting hue angle*, *ending hue angle*], where the given coordinates represent a counter clockwise direction. Given the index coordinate of the current job, Figure 4.6 (a) shows the limit at which the next color can be located without setup based on the matrix in Figure 4.5. This space is denoted by D_H [*min value* (H_c-1), *max value* (H_c +1)]. Figure 4.6 (b) displays the top view of the hue space. If H_c-1 is below 1, then set H_c-1 to 8. If H_c +1 is above 8, then set H_c +1 to 1.



(a) Hue setup matrix space

(b) Top view of the hue space

Figure 4.6. Hue space

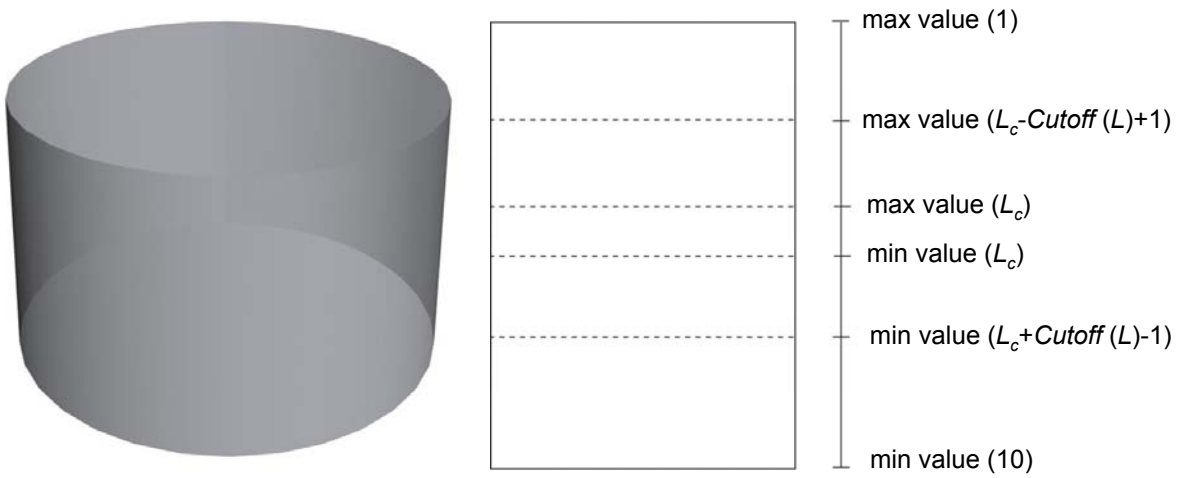
4.3.3. Setup Matrix by Lightness Index

Figure 4.7 shows the setup time matrix when setup is needed due to the difference between two lightness indices. Since boiling out time still depends on the dyeing profile of the fabric/dye type, the setup time is determined by the current fabric/dye type. However, these setup times are sequence dependent in that no setup is required if a job with the same or higher index is processed following the current job on the machine. The cleaning process should be inserted when white (index 1) follows any other color or when black (index 10) is followed by any other color. Otherwise the setup is inserted if the current lightness index is greater than the following lightness index by a specified value, C*utoff* (*L*), or more. C*utoff* (*L*) equals 3 for index 2 to 9, and C*utoff* (*L*) equals 1 for index 1 and index 10. This number is based on industry practice.

F\T	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	s_F	0	0	0	0	0	0	0	0	0
3	s_F	0	0	0	0	0	0	0	0	0
4	s_F	0	0	0	0	0	0	0	0	0
5	s_F	s_F	0	0	0	0	0	0	0	0
6	s_F	s_F	s_F	0	0	0	0	0	0	0
7	s_F	s_F	s_F	s_F	0	0	0	0	0	0
8	s_F	s_F	s_F	s_F	s_F	0	0	0	0	0
9	s_F	s_F	s_F	s_F	s_F	s_F	0	0	0	0
10	s_F	s_F	s_F	s_F	s_F	s_F	s_F	s_F	s_F	0

Figure 4.7. Lightness setup matrix

Let lightness space be defined as the color space which covers all the hue indices, certain lightness indices and all the chroma indices. It will be denoted by D_L [*starting lightness value*, *ending lightness value*]. Given the index coordinate of the current job, Figure 4.8 (a) shows the limit at which the next color can be located without setup based on the matrix in Figure 4.7. This space is denoted by D_L [*min value* (10), *max value* (L_c - *Cutoff* (L)+1)]. Figure 4.8 (b) displays a side view of the lightness space. If L_c - *Cutoff* (L) +1 is below 2, then set L_c - *Cutoff* (L) +1 to 2 unless L_c = 1. If L_c + *Cutoff* (L) -1 is above 10, then set L_c + *Cutoff* (L) -1 to 10.



(a) Lightness setup matrix space (b) Side view of the lightness space

Figure 4.8. Lightness space

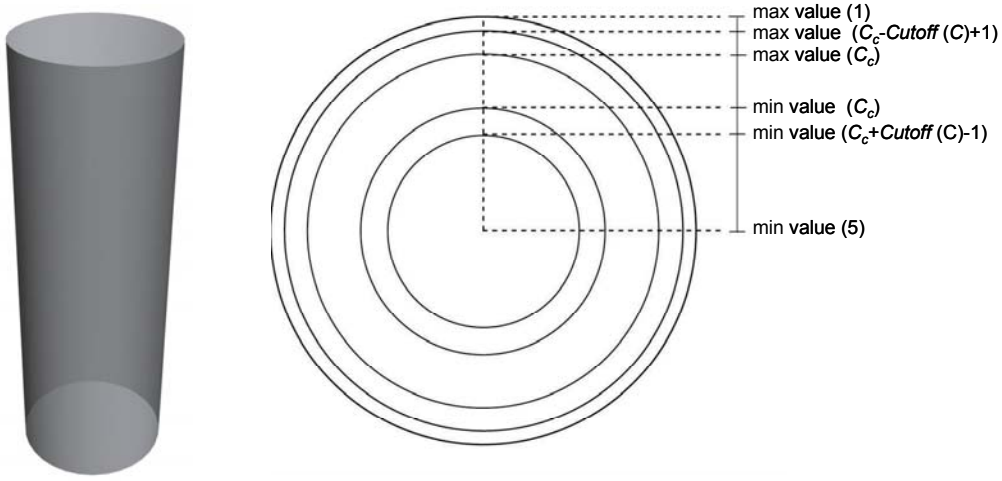
4.3.4. Setup Matrix by Chroma Index

Figure 4.9 shows the setup time matrix when setup is needed due to the difference between two chroma indices. Setup is not needed if a job with the same or higher index is processed following the current job on the machine. Setup is inserted if the current chroma index is greater than following chroma index by a specified value, C*utoff* (*C*), or more. In this example C*utoff* (*C*) equals 2 for all indices.

F\T	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	s_F	0	0	0	0
4	s_F	s_F	0	0	0
5	s_F	s_F	s_F	0	0

Figure 4.9. Chroma setup matrix

Let chroma space be defined as the color space which covers all the hue indices, all the lightness indices and certain chroma indices. It will be denoted by D_C [*starting chroma value*, *ending chroma value*]. Given the index coordinate of the current job, Figure 4.10 (a) shows the limit at which the next color can be located without setup based on the matrix in Figure 4.9. This space is denoted by D_C [*min value* (5), *max value* (C_c - *Cutoff* (C)+1)]. Figure 4.10 (b) displays a top view of the chroma space. If C_c - *Cutoff* (C) +1 is below 1, then set C_c - *Cutoff* (C) +1 to 1. If C_c + *Cutoff* (C) -1 is above 5, then set C_c + *Cutoff* (C) -1 to 5.



(a) Chroma setup matrix space (b) Top view of the chroma space

Figure 4.10. Chroma space

4.3.5. Interdependency between Setup Matrices

Matrices will now be developed to capture the interdependency between the hue, lightness and chroma indices. First we treat the interdependency between the hue and lightness index. If the lightness of the preceding color is much lighter than that of current color, meaning if the lightness index is decreased by the *Cutoff* (L) from L_c, setup should be inserted. Given an index coordinate of a job, the limit at which next color can be located

without setup can be represented by taking the intersection of the hue space and the lightness space as follows:

$$D_H\,[\textit{min value}\,(H_c\text{-}1),\ \textit{max value}\,(H_c\text{+}1)]$$

$$\cap\ D_L\,[\textit{min value}\,(10),\ \textit{max value}\,(L_c\text{-}\ \textit{Cutoff}\,(L)\text{+}1)]$$

However, if the lightness of the preceding color is much darker than that of the current color, in other words if the lightness index is increased by at least *Cutoff* (*L*), setup is not necessary regardless of hue index. Therefore the complete setup-free zone induced by the hue and lightness indices can be expressed as follows:

$$\{D_H\,[\textit{min value}\,(H_c\text{-}1),\ \textit{max value}\,(H_c\text{+}1)]$$

$$\cap\ D_L\,[\textit{min value}\,(10),\ \textit{max value}\,(L_c\text{-}\ \textit{Cutoff}\,(L)\text{+}1)]\}$$

$$\cup\ D_L\,[\textit{min value}\,(10),\ \textit{min value}\,(L_c\text{+}\ \textit{Cutoff}\,(L)\text{-}1)]$$

This space is shown in Figure 4.11.

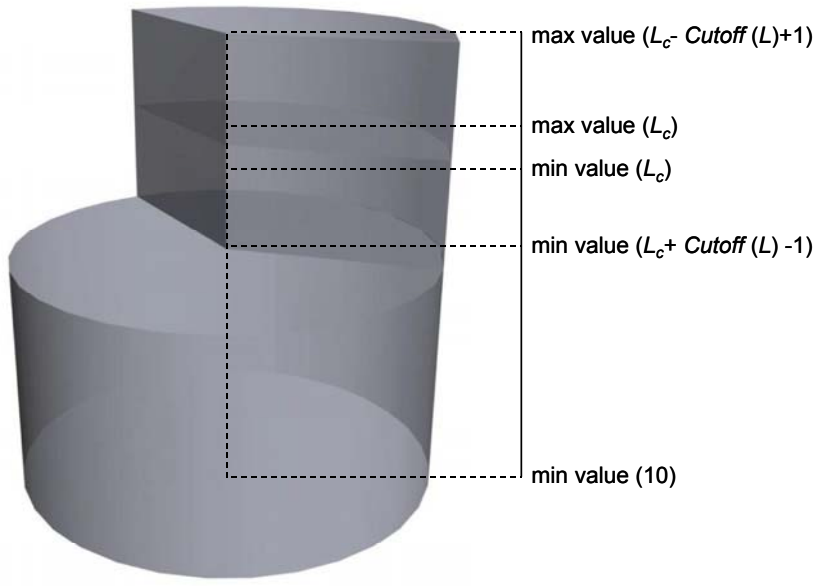


Figure 4.11. Setup-free zone induced by hue and lightness index

Figure 4.12 shows the expanded setup matrix induced by the combination of the hue and lightness indices. The pink block represents that there is no need of setup insertion due to the hue index difference of the current and preceding index coordinates.

$$Pink\ block = D_H\,[min\ value\ (H_c\text{-}1),\ max\ value\ (H_c\text{+}1)]$$
$$\bigcap D_L\,[min\ value\ (10),\ max\ value\ (L_c\text{-}\ Cutoff\ (L)\text{+}1)]$$

However, if the lightness index is decreased by at least *Cutoff* (*L*) from L_c, setup should be inserted. The violet block represents these setups.

$$Violet\ block = D_H\,[min\ value\ (H_c\text{-}1),\ max\ value\ (H_c\text{+}1)]$$
$$\bigcap D_L\,[max\ value\ (L_c\text{-}\ Cutoff\ (L)\text{+}1),\ max\ value\ (1)]$$

Even though setup should be inserted due to the hue difference alone, if the lightness index is increased by at least *Cutoff* (*L*), setup is not necessary regardless of hue index. The yellow block represents these canceled setups.

$$Yellow\ block = D_H\,[max\ value\ (H_c\text{+}1),\ min\ value\ (H_c\text{-}1)]$$
$$\bigcap D_L\,[min\ value\ (10),\ min\ value\ (L_c\text{+}\ Cutoff\ (L)\text{-}1)]$$

The green block represents the setup insertion due to the hue index difference.

$$Green\ block = D_H\,[max\ value\ (H_c\text{+}1),\ min\ value\ (H_c\text{-}1)]$$
$$\bigcap D_L\,[min\ value\ (L_c\text{+}\ Cutoff\ (L)\text{-}1),\ max\ value\ (1)]$$

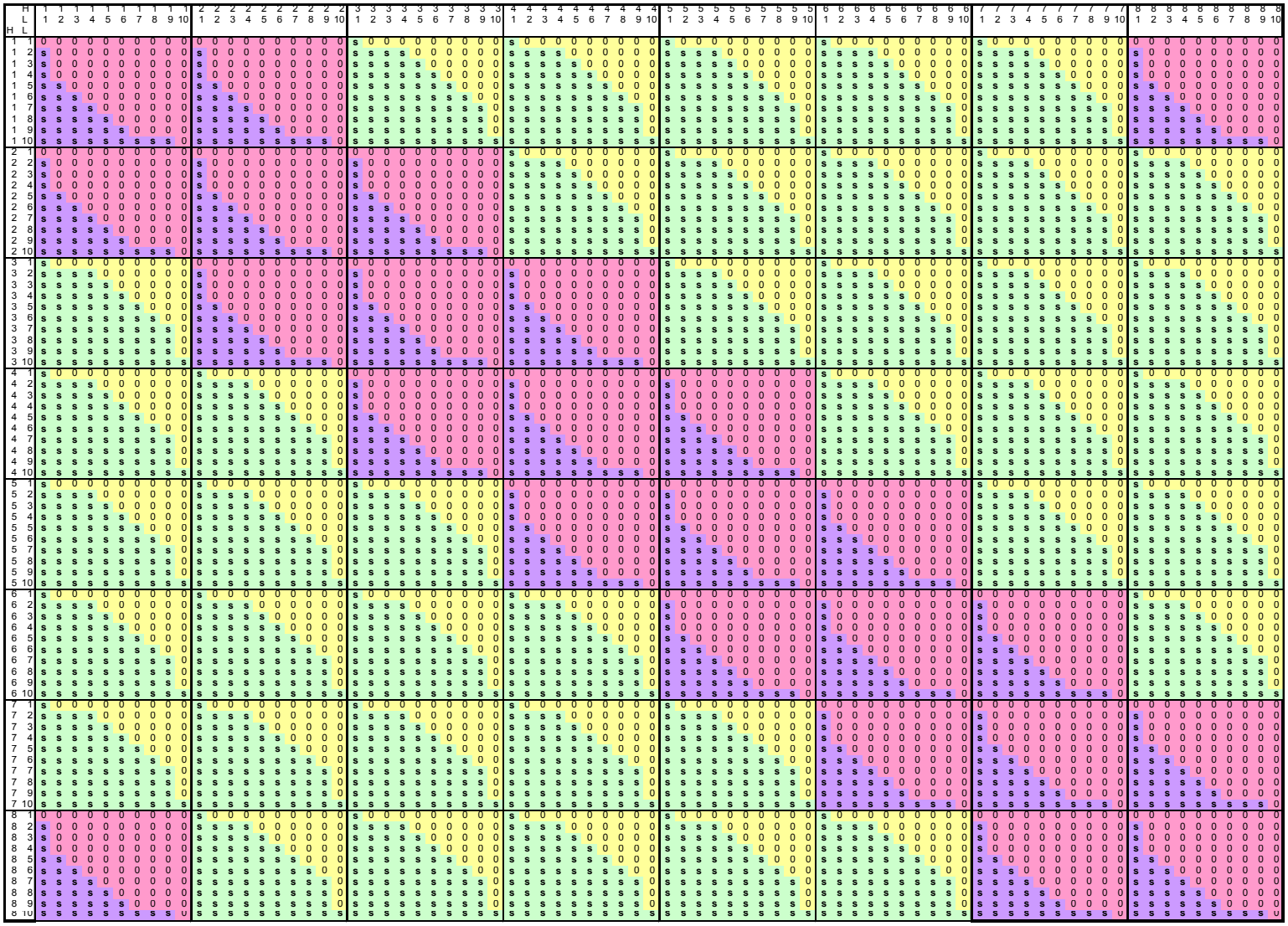


Figure 4.12. Expanded setup matrix induced by the hue and lightness indices

Next we consider the effect of the chroma on the setup-free zone induced by the hue and lightness indices. If the chroma of the preceding color is much stronger than that of current color, in other words if chroma index is decreased by *Cutoff* (*C*), setup should be inserted. Following this logic, the setup-free zone induced by the hue and lightness indices in Figure 4.11 is reshaped by peeling off the outer space corresponding to the inserted setups (Figure 4.13). This zone can be expressed as follows:

$$\{\{D_H\,[\textit{min value}\,(H_c\text{-}1),\ \textit{max value}\,(H_c\text{+}1)]$$
$$\cap\ D_L\,[\textit{min value}\,(10),\ \textit{max value}\,(L_c\text{-}\ \textit{Cutoff}\,(L)\text{+}1)]\}$$
$$\cup\ D_L\,[\textit{min value}\,(10),\ \textit{min value}\,(L_c\text{+}\ \textit{Cutoff}\,(L)\text{-}1)]\}$$
$$-\{D_H\,[\textit{min value}\,(H_c\text{-}1),\ \textit{max value}\,(H_c\text{+}1)]$$
$$\cap D_L\,[\textit{min value}\,(L_c\text{+}\ \textit{Cutoff}\,(L)\text{-}1),\ \textit{max value}\,(L_c\text{-}\ \textit{Cutoff}\,(L)\text{+}1)]$$
$$\cap D_C\,[\textit{max value}\,(C_c\text{-}\ \textit{Cutoff}\,(C)\text{+}1),\ \textit{max value}\,(1)]\}$$

Alternatively, this space also can be expressed as follows:

$$\{D_H\,[\textit{min value}\,(H_c\text{-}1),\ \textit{max value}\,(H_c\text{+}1)]$$
$$\cap\ D_L\,[\textit{min value}\,(L_c\text{+}\ \textit{Cutoff}\,(L)\text{-}1),\ \textit{max value}\,(L_c\text{-}\ \textit{Cutoff}\,(L)\text{+}1)]$$
$$\cap\ D_C\,[\textit{min value}\,(5),\ \textit{max value}\,(C_c\text{-}\ \textit{Cutoff}\,(C)\text{+}1)]\}$$
$$\cup D_L\,[\textit{min value}\,(10),\ \textit{min value}\,(L_c\text{+}\ \textit{Cutoff}\,(L)\text{-}1)]$$

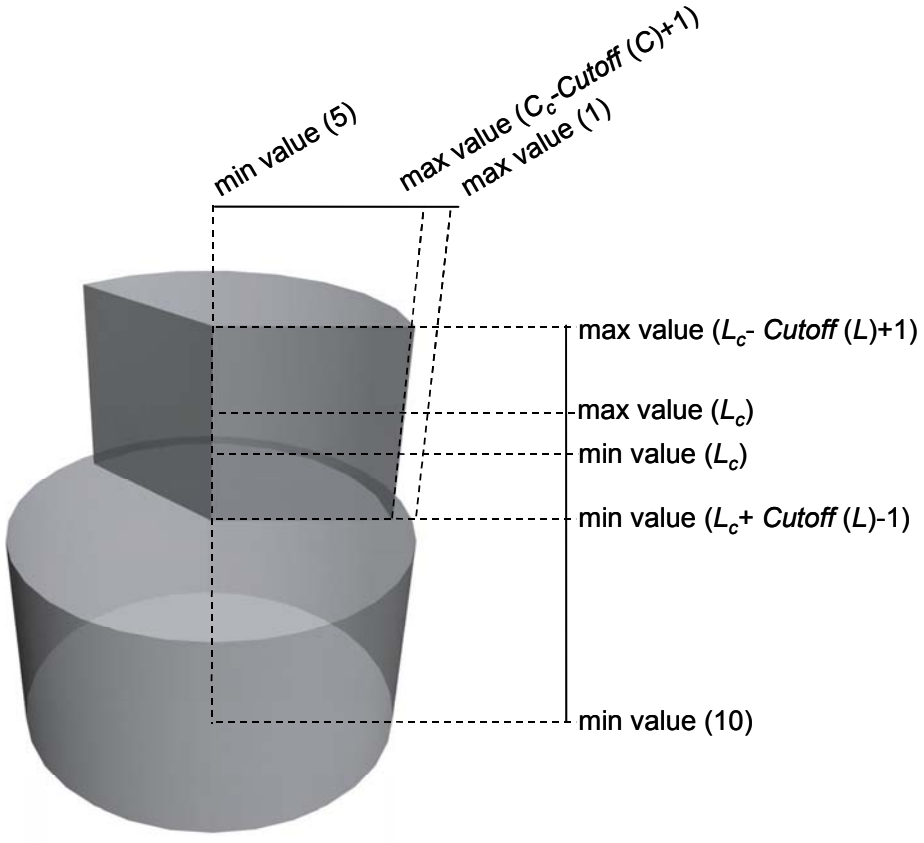


Figure 4.13. Setup-free zone modification decreased by *Cutoff* (*C*)

The setup matrix in Figure 4.12 can be further extended to include the chroma index. In Figure 4.14, the pink and violet blocks of the setup matrix in Figure 4.12 are expanded to include the chroma index. Basically, each 1×1 matrix in Figure 4.12 expands to a 5×5 matrix in Figure 4.14. When the lightness difference between the current and preceding color is not large enough, corresponding to D_L [*min value* (L_c+ *Cutoff* (*L*)-1), *max value* (L_c- *Cutoff* (*L*)+1)], the chroma difference is important in deciding whether to insert setup or not. If the chroma index is decreased by at least *Cutoff* (*C*) from C_c, setup should be inserted. The dark blue block in Figure 4.14 represents these inserted setups.

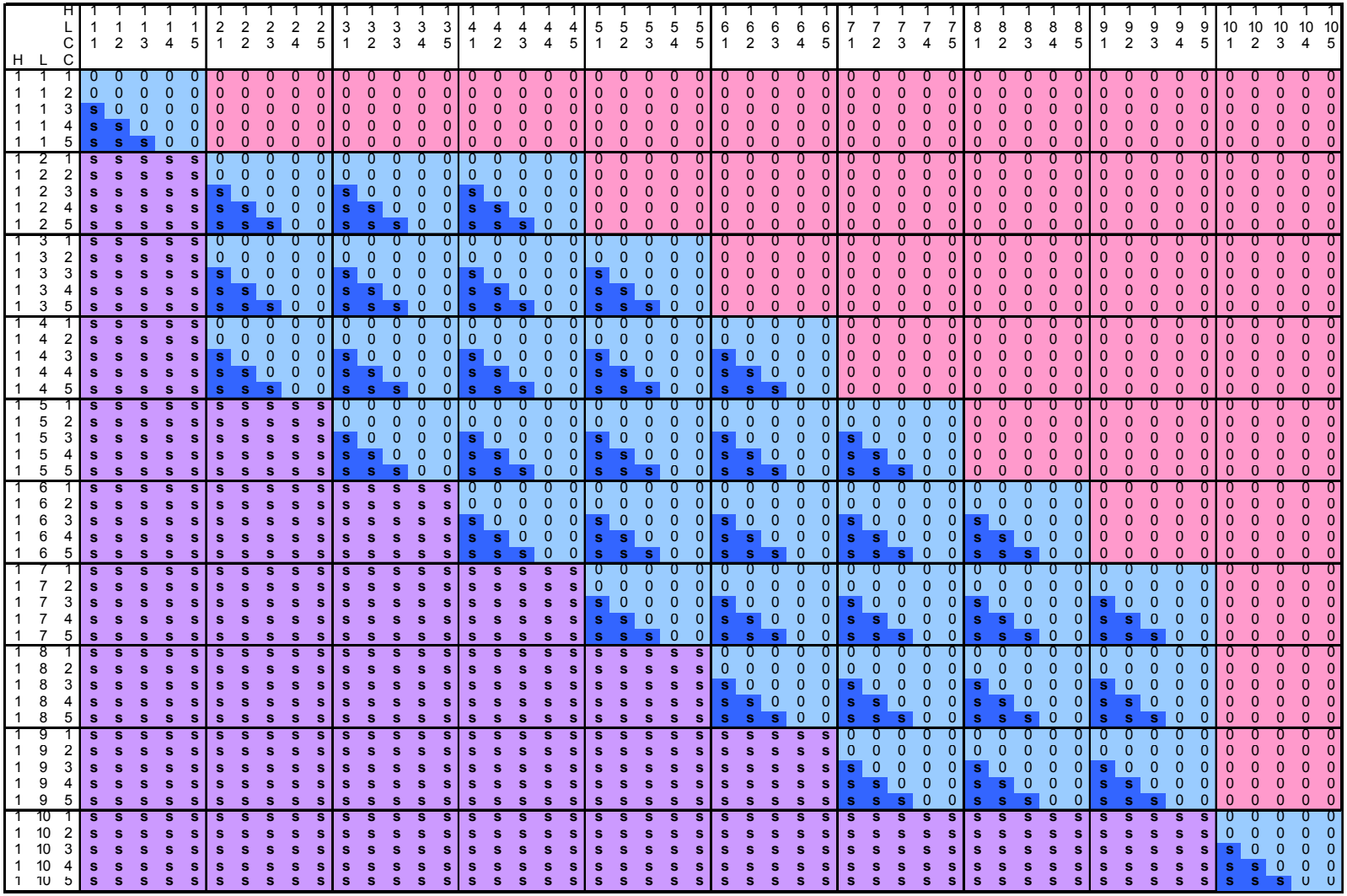


Figure 4.14. Expanded pink and violet blocks of setup matrix in Figure 4.12

$$Pink\ block = D_H\,[min\ value\ (H_c\text{-}1),\ max\ value\ (H_c\text{+}1)]$$
$$\cap\, D_L\,[min\ value\ (10),\ min\ value\ (L_c\text{+}\ Cutoff\ (L)\text{-}1)]$$

$$Light\ Blue\ block = D_H\,[min\ value\ (H_c\text{-}1),\ max\ value\ (H_c\text{+}1)]$$
$$\cap\ D_L\,[min\ value\ (L_c\text{+}\ Cutoff\ (L)\text{-}1),\ max\ value\ (L_c\text{-}\ Cutoff\ (L)\text{+}1)]$$
$$\cap\ D_C\,[min\ value\ (5),\ min\ value\ (C_c\text{-}\ Cutoff\ (C))]$$

$$Dark\ blue\ block = D_H\,[min\ value\ (H_c\text{-}1),\ max\ value\ (H_c\text{+}1)]$$
$$\cap\ D_L\,[min\ value\ (L_c\text{+}\ Cutoff\ (L)\text{-}1),\ max\ value\ (L_c\text{-}\ Cutoff\ (L)\text{+}1)]$$
$$\cap\ D_C\,[max\ value\ (C_c\text{-}\ Cutoff\ (C)\text{+}1),\ max\ value\ (1)]$$

$$Violet\ block = D_H\,[min\ value\ (H_c\text{-}1),\ max\ value\ (H_c\text{+}1)]$$
$$\cap\ D_L\,[max\ value\ (L_c\text{-}\ Cutoff\ (L)\text{+}1),\ max\ value\ (1)]$$

However, if the chroma of the preceding color is much duller than that of the current color, in other words if chroma index is increased by at least *Cutoff* (*C*), setup is not necessary regardless of hue index. Following this logic, the setup-free zone in Figure 4.13 is reshaped by adding the core chroma space corresponding canceled setups (Figure 4.15). The modified setup-free zone can be expressed as follows:

$$\{\{\{D_H\,[min\ value\ (H_c\text{-}1),\ max\ value\ (H_c\text{+}1)]$$
$$\cap\ D_L\,[min\ value\ (10),\ max\ value\ (L_c\text{-}\ Cutoff\ (L)\text{+}1)]\}$$
$$\cup\ D_L\,[min\ value\ (10),\ min\ value\ (L_c\text{+}\ Cutoff\ (L)\text{-}1)]\}$$
$$-\{D_H\,[min\ value\ (H_c\text{-}1),\ max\ value\ (H_c\text{+}1)]$$

$\cap D_L$ [*min value* (L_c+ *Cutoff* (L)-1), *max value* (L_c- *Cutoff* (L)+1)]

$\cap D_C$ [*max value* (C_c- *Cutoff* (C)+1), *max value* (1)]}

$\cup$ {D_L [*min value* (L_c+ *Cutoff* (L)-1), *max value* (L_c- *Cutoff* (L)+1)]

$\cap$ D_C [*min value* (5), *min value* (C_c+ *Cutoff* (C)-1)]}

Alternatively this space also can be expressed as follows:

{D_H [*min value* (H_c-1), *max value* (H_c+1)]

$\cap$ D_L [*min value* (L_c+ *Cutoff* (L)-1), *max value* (L_c- *Cutoff* (L)+1)]

$\cap$ D_C [*min value* (*5*), *max value* (C_c- *Cutoff* (C)+1)]}

$\cup$ {D_L [*min value* (L_c+ *Cutoff* (L)-1), *max value* (L_c- *Cutoff* (L)+1)]

$\cap$ D_C [*min value* (*5*), *min value* (C_c+ *Cutoff* (C)-1)]}

$\cup$ D_L [*min value* (10), *min value* (L_c+ *Cutoff* (L)-1)]

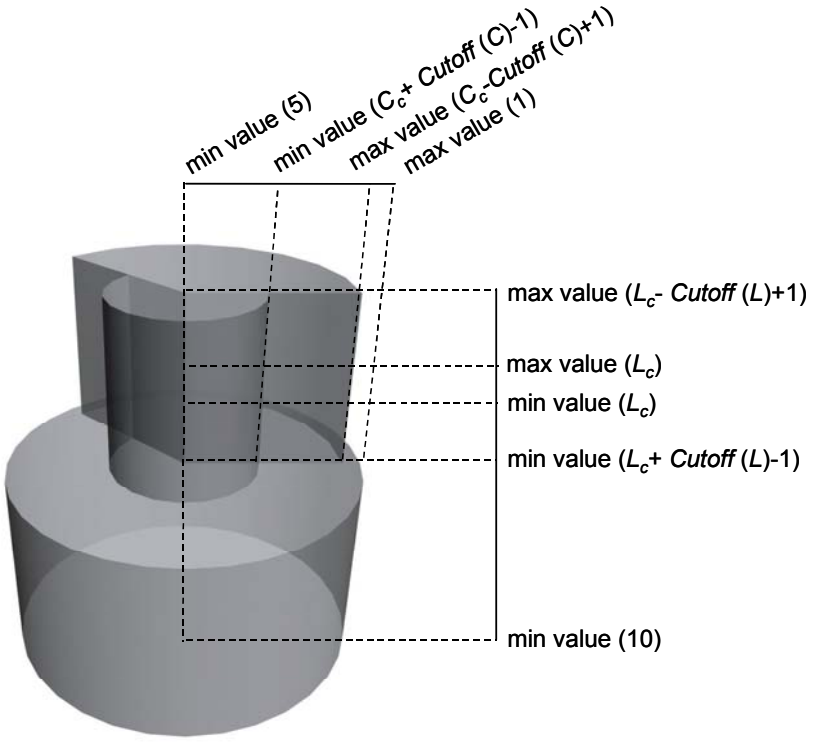


Figure 4.15. Setup-free zone induced by hue, lightness and chroma

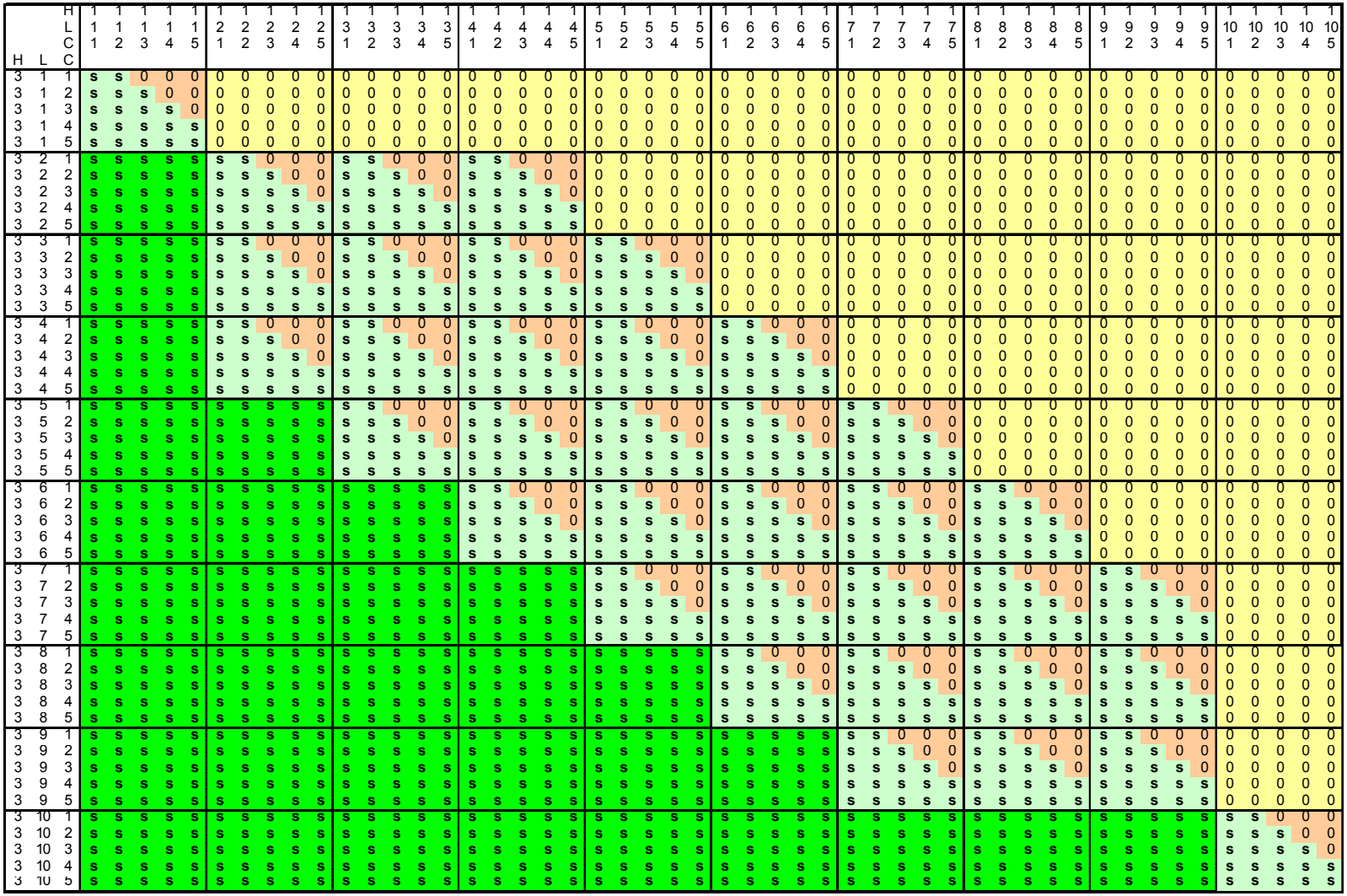
In Figure 4.16, the green and yellow blocks of the setup matrix in Figure 4.12 are expanded to include the chroma index. If the chroma index is increased by at least *Cutoff* (*C*), setup is not necessary regardless of hue index. The orange block in Figure 4.16 represents these canceled setups.

$$\textit{Yellow block} = D_H\,[\textit{max value}\,(H_c{+}1),\ \textit{min value}\,(H_c{-}1)]$$
$$\cap\ D_L\,[\textit{min value}\,(10),\ \textit{min value}\,(L_c{+}\ \textit{Cutoff}\,(L){-}1)]$$

$$\textit{Orange block} = D_H\,[\textit{max value}\,(H_c{+}1),\ \textit{min value}\,(H_c{-}1)]$$
$$\cap\ D_L\,[\textit{min value}\,(L_c{+}\ \textit{Cutoff}\,(L){-}1),\ \textit{max value}\,(L_c{-}\ \textit{Cutoff}\,(L){+}1)]$$
$$\cap\ D_C\,[\textit{min value}\,(5),\ \textit{min value}\,(C_c{+}\ \textit{Cutoff}\,(C){-}1)]$$

$$\textit{Yellow-green block} = D_H\,[\textit{max value}\,(H_c{+}1),\ \textit{min value}\,(H_c{-}1)]$$
$$\cap\ D_L\,[\textit{min value}\,(L_c{+}\ \textit{Cutoff}\,(L){-}1),\ \textit{max value}\,(L_c{-}\ \textit{Cutoff}\,(L){+}1)]$$
$$\cap\ D_C\,[\textit{min value}\,(C_c{+}\ \textit{Cutoff}\,(C){-}1),\ \textit{max value}\,(1)]$$

$$\textit{Green block} = D_H\,[\textit{max value}\,(H_c{+}1),\ \textit{min value}\,(H_c{-}1)]$$
$$\cap D_L\,[\textit{max value}\,(L_c{-}\ \textit{Cutoff}\,(L){+}1),\ \textit{max value}\,(1)]$$



H	L	HLCC	1 1 1	1 1 2	1 1 3	1 1 4	1 1 5	1 2 1	1 2 2	1 2 3	1 2 4	1 2 5	1 3 1	1 3 2	1 3 3	1 3 4	1 3 5	1 4 1	1 4 2	1 4 3	1 4 4	1 4 5	1 5 1	1 5 2	1 5 3	1 5 4	1 5 5	1 6 1	1 6 2	1 6 3	1 6 4	1 6 5	1 7 1	1 7 2	1 7 3	1 7 4	1 7 5	1 8 1	1 8 2	1 8 3	1 8 4	1 8 5	1 9 1	1 9 2	1 9 3	1 9 4	1 9 5	1 10 1	1 10 2	1 10 3	1 10 4	1 10 5
3	1	1	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	2	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	3	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	4	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	5	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	1	s	s	s	s	s	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	2	s	s	s	s	s	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	3	s	s	s	s	s	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	4	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	5	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	1	s	s	s	s	s	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	2	s	s	s	s	s	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	3	s	s	s	s	s	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	4	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	5	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	1	s	s	s	s	s	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	2	s	s	s	s	s	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	3	s	s	s	s	s	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	4	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	5	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	5	1	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	5	2	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	5	3	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	5	4	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	5	5	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	6	1	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	6	2	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	6	3	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	6	4	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	6	5	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0
3	7	1	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0
3	7	2	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0
3	7	3	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	s	s	s	s	0	0	0	0	0	0	0	0	0	0	0	0
3	7	4	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0
3	7	5	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0
3	8	1	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0
3	8	2	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0
3	8	3	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0	0
3	8	4	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0	0
3	8	5	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	0	0	0	0	0
3	9	1	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	s	s	0	0	0	s	s	0	0	0	0	0	0	0	0	0	0	0	0	0
3	9	2	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0	0	s	s	s	0	0	s	s	s	0	0	0	0	0	0	0	0	0	0	0
3	9	3	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0	0
3	9	4	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	0
3	9	5	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
3	10	1	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
3	10	2	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
3	10	3	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
3	10	4	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
3	10	5	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s

Figure 4.16. Expanded green and yellow blocks of setup matrix in Figure 4.12

4.4. Model Validation

There is a concern as to whether the model of the cylindrical color space represents the true color space. Based on this concern, several object-color systems were investigated, which are systems of object-color samples organized according to colorant characteristics, color-stimulus syntheses, or color response [Burnham, R. W. et al., 1963]. The closest one to the model in this dissertation is the *Munsell Color System* which was developed from judgments of equal hue, value (brightness or lightness), and chroma (saturation). This includes a scale of 40 hues and 9 values. (Figure 4.17).

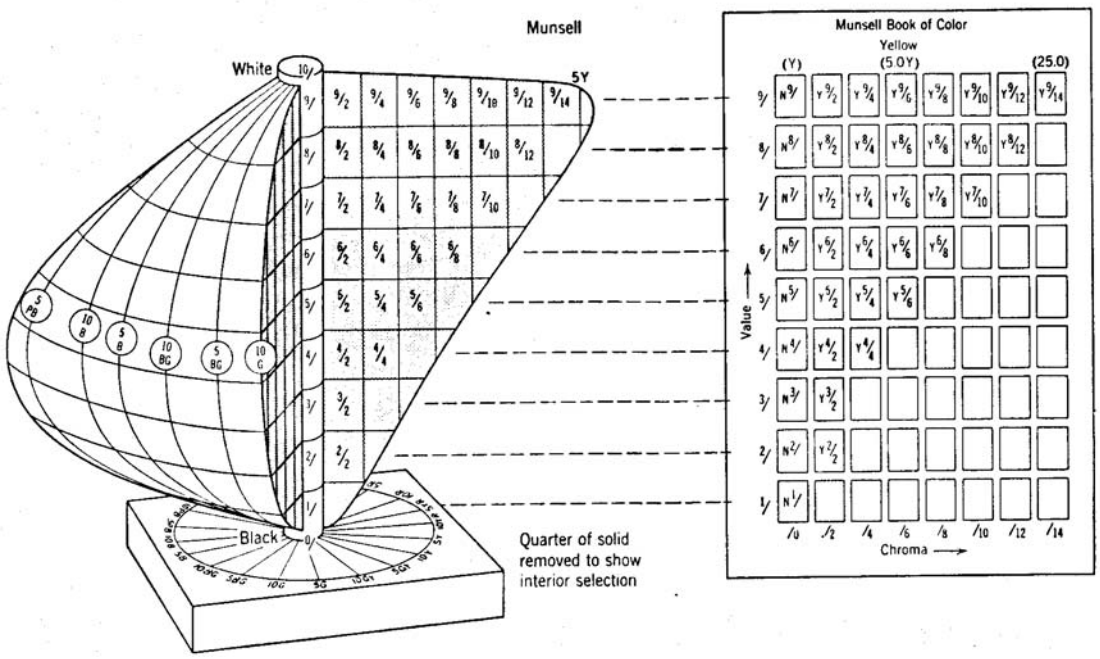


Figure 4.17. Munsell Color System [Burnham, R. W. et. al., 1963]

Based on reviewing the *Munsell Color System*, a discrepancy between it and the cylindrical model was found. Certain regions in the cylindrical model are not present in the *Munsell Color System*. For example, according to Figure 4.17, the colors with low value and high

chroma do not exist in the yellow region. However, since the cylindrical model covers a larger space than the true color space from the *Munsell Color System*, further applications based on the cylindrical model also can be applied to the true color space. If the jobs with a color in the cylindrical model can be scheduled, then the jobs with a color in the true color space also can be scheduled. Future research may refine the cylindrical color space model based on the object-color system.

CHAPTER 5. SCHEDULING BATCHWISE FABRIC DYEING OPERATIONS

5.1. Introduction

In order to minimize the number of machine cleanings, the ideal sequence is to run the same color repeatedly on a particular machine. For this reason, dyeing plants have more than one dyeing machine so that they can process different types of orders on different machines. However, this can cause under-utilization of machines due to unoptimized scheduling. According to Baker [1999], the single machine model provides us with an opportunity to discover heuristic rules, to test conjectures, and to develop working principles that may generalize to more complicated models. Moreover, single machine models are also relevant when there is effectively one stage of production or when the production process is dominated by a single bottleneck machine.

In this chapter, experiments with a batchwise dyeing operation are performed based on a modification of Taner's algorithm by incorporating the setup time matrix discussed in Chapter 4. The one machine dyeing problem will be considered when we develop heuristics to schedule dyeing operations. However, the setup matrices developed in Chapter 4 are valid regardless of the number of dyeing machines. Flowshop scheduling problems will be treated that focus on Taner's problem generation method in which all jobs must go through one sequence dependent machine, occurring in the middle of their route. This situation would occur when fabric is prepared, dyed and finished in one factory. Even though preparation and finishing operations may also have sequence dependent setups, in this chapter it is assumed that only the dyeing operation has sequence dependent setups.

5.2. Modified Scheduling Algorithm (Algorithm 1)

Meeting customer due dates is an important objective as well as minimizing the number of setups. Recall that Taner used three sequencing rules, SST, SUBSET and EDD, in his scheduling algorithm for the job shop problem (Refer to 2.3.2). If all jobs in the queue were already tardy and at least one was extremely tardy, the SUBSET rule was used. If the advantage from minimizing the number of setups did not exceed that of processing the job which potentially yields the maximum lateness, the EDD rule was used to give higher priority to the job which potentially yields the maximum lateness. If minimizing the setup was more important, then the SST rule was used. Sequencing jobs in revised slack order was considered equivalent to using the EDD rule.

Taking into account the new index format of each job, a new SST rule for dyeing operations (SST_D) was developed and will be explained further in the following section. The EDD and SUBSET rules are essentially unchanged. The only change in the SUBSET rule is that the SST_D rule will be used within the priority groups instead of Taner's SST rule. The decision flow of the modified version of Taner's algorithm, Algorithm 1, is given as follows:

Algorithm 1

Step 1. Add the jobs in queue to the consideration list. If this is not the first iteration, also add jobs which are estimated to arrive in the next *T* time units to the consideration list.

Step 2. If no jobs are on the consideration list, do nothing.

Step 3. If there is one job on the list, then choose the job on the list.

Step 4. If there is more than one job, determine which rule is preferred.

4.1. If all the jobs waiting in the queue are already tardy and at least one is tardy by γ or more time units, choose the *SUBSET job*.

4.2. If the SUBSET rule is not chosen, find both the jobs selected by the EDD rule and SST_D rule and determine whether the *EDD job* or *SST job* should be chosen as follows:

- If the *SST job* has τ time units or more until its due date while the *EDD job* is more urgent, choose the *EDD job*.
- Otherwise, choose the *SST job*.

Step 5. If the chosen job is already in the queue, select it. If not, try to find the job in the queue which can be finished by the time the chosen job arrives and finishes its setup operation.

5.2.1. Shortest Setup Time for Dyeing Operation (SST_D)

In industry, dyeing operations are often scheduled with the primary goal of minimizing setups. Jobs are usually first categorized based on their fabric/dye type and batches are formed. Setup is inserted whenever the fabric/dye type is changed, and this setup is independent of the hue, lightness, and chroma of the color being dyed. Then, jobs within the same fabric/dye type are sequenced by hue, lightness, and chroma indices. Setup is inserted based on the setup matrix developed in Chapter 4. When meeting customer delivery dates is the primary objective, these two procedures are interrelated by due date consideration. Therefore, it is often advantageous to process jobs of the same fabric/dye type in more than one batch. Sequencing jobs within the same fabric/dye type is a complex problem itself, and the interdependency between scheduling jobs within a batch and scheduling those batches

makes it even more complex. Therefore, as a first step, an algorithm for sequencing jobs within the same fabric/dye type is developed in this dissertation, and the fabric/dye type consideration is excluded from the SST_D rule. In other words, the objective of the SST_D rule is to find the longest possible sequence of jobs before a setup is required within a given fabric/dye type.

Figure 5.1 depicts the decision flow of the SST_D rule and the following definitions are required for this rule:

- Q - The set of jobs in the queue
- R - The set of jobs in the queue which have not yet been considered
- N_i -The number of jobs in $Q - \{i\}$, $i \in Q$, which do not require a setup if they are immediately processed after job i
- α - The job j which corresponds to the maximum value of N_j among all jobs j which do not require a setup if they are processed after the current job on the machine
- β - The job k which corresponds to the maximum value of N_k among all jobs k which do require a setup if they are processed after the current job on the machine

Let the current job on the machine have the three indices $\{H_c, L_c, C_c\}$. Also let job i have the three indices $\{H_i, L_i, C_i\}$.

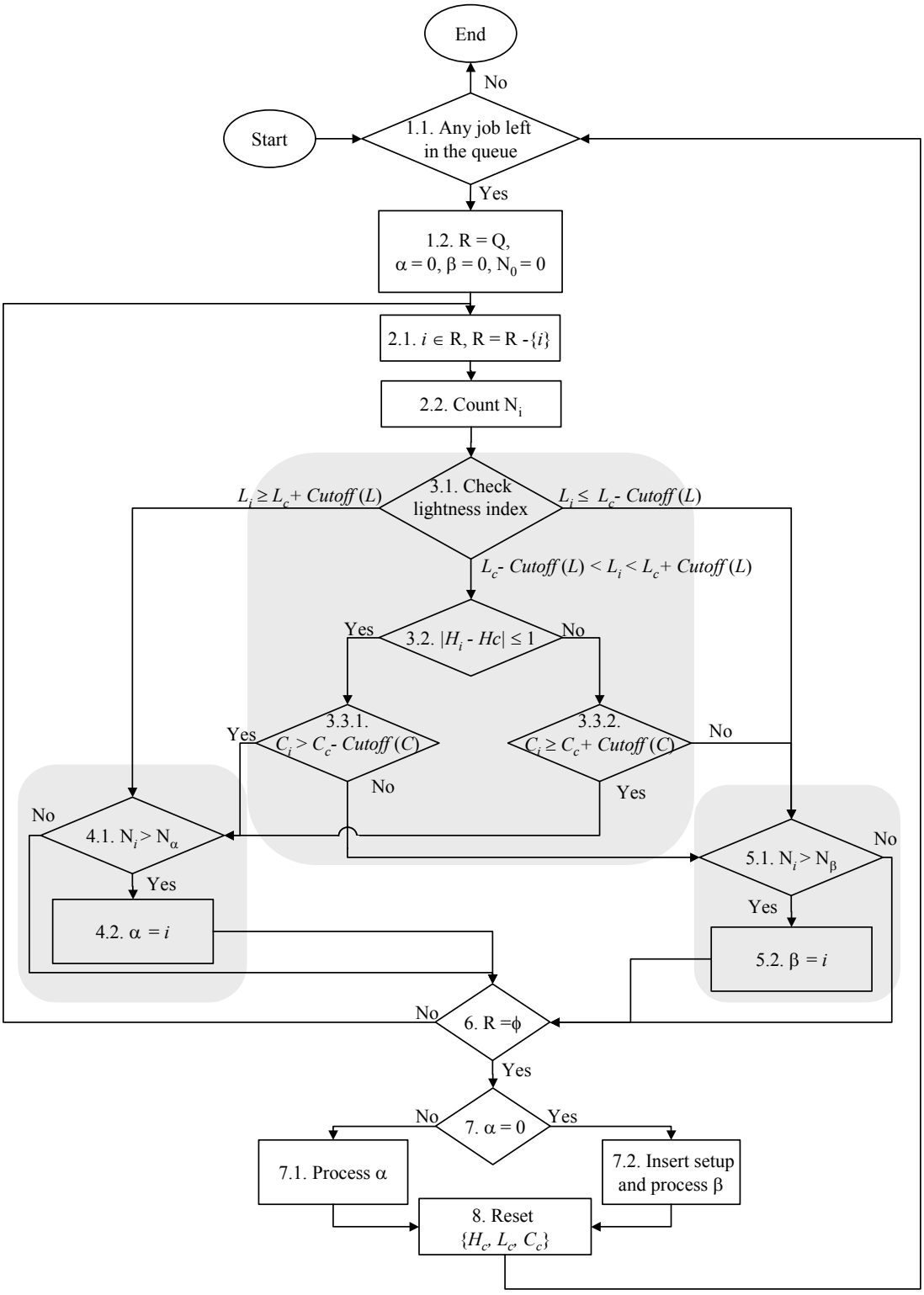


Figure 5.1. The decision flow of SST_D rule

The basic idea is to try to find a job that (i) does not require a setup before it can be processed and (ii) will allow the maximum number of jobs to be processed after it before a setup is necessary. Condition (i) is given priority over (ii) to postpone initiating a setup until there is no other choice. In other words, a job that does not require a setup to be processed after the current job on the machine will always be chosen before a job that requires one, regardless of the number of jobs that can be processed afterward without a setup. Since finding a job that satisfies (ii) would require a lot of computational time, a simple heuristic is used to try to achieve this goal. This heuristic is to look at each job in the queue and count the number of other jobs in the queue that could be processed immediately after it without setup. Table 5.1 shows an example of this job counting procedure.

Table 5.1. An example of the job counting procedure

(a) Table of setup insertion after the current job

	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6
Is the job in the setup-free zone of current job?	Yes	No	No	Yes	No	Yes

(b) Matrix of setup insertion between all the jobs in the queue

	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6	Job count N_i
Is the job in setup-free zone of job 1		Yes	Yes	No	No	No	2
Is the job in setup-free zone of job 2	No		No	No	No	Yes	1
Is the job in setup-free zone of job 3	Yes	No		Yes	Yes	Yes	4
Is the job in setup-free zone of job 4	Yes	Yes	No		Yes	No	3
Is the job in setup-free zone of job 5	No	Yes	No	Yes		Yes	3
Is the job in setup-free zone of job 6	Yes	No	Yes	Yes	Yes		4

Suppose there are 6 jobs in the queue. Since jobs 1, 4, and 6 do not require setup when they come after the current job according to Table 5.1 (a), they become a candidate for the next job processed. For job 4, there are 3 jobs in its setup-free zone among the jobs in the queue. Thus the job count for job 4 is 3. Among jobs 1, 4, and 6, the job with the largest job count will be chosen to be the next job processed. In this example, job 6 is chosen (i.e., α = job 6).

This heuristic assumes that if jobs could be processed immediately after a given job without a setup, then there is likely to be a way to process these jobs in some sequence without any setups. This assumption is not always true. For example, if the current job has index coordinates of {3, 1, 2} and there are two jobs in the queue which have index coordinates of {2, 3, 2} and {4, 4, 2}, then the assumption is violated. Although these two jobs are located inside of the setup-free zone of the current job, a setup is required between the two jobs due to the hue index difference, if they are processed consecutively after the current job.

SST_D Rule

1. Check the queue and set initial values if necessary.
   1.1. If a job is left in the queue, go to step 1.2. Otherwise end.
   1.2. Set R = Q, α = 0, β = 0 and N_0 = 0.
2. Select job *i* from set R
   2.1. Set R = R – {*i*}.
   2.2. Count the number of jobs in the setup-free zone given by the index coordinates of job *i*

3. Check whether or not job *i* is located in the setup-free zone given by the index coordinate of the current job on the machine.

   3.1. Check the lightness index of job *i*.

      3.1.1. If $L_i \geq L_c + Cutoff\ (L)$, go to step 4.

      3.1.2. If $L_c - Cutoff\ (L) < L_i < L_c + Cutoff\ (L)$, go to step 3.2.

      3.1.3. Go to step 5.

   3.2. Check the hue index of job *i*. If $|H_i - H_c| \leq 1$ then go to step 3.3.1. Otherwise go to step 3.3.2.

   3.3. Check the chroma index of job *i*.

      3.3.1. If $C_i > C_c - Cutoff$, then go to step 4. Otherwise go to step 5.

      3.3.2. If $C_i \geq C_c + Cutoff$, then go to step 4. Otherwise go to step 5.

4. Update job α.

   4.1. If N_i is not greater than N_α, go to step 6.

   4.2. Set α equal to *i* and go to step 6.

5. Update job β.

   5.1. If N_i is not greater than N_β, go to step 6.

   5.2. Set β equal to *i* and go to step 6.

6. If set R is empty, go to step 7. Otherwise go to step 2.

7. Check whether α exists or not. If α = 0, go to step 7.2. Otherwise go to step 7.1.

   7.1. Process α and go to step 8

   7.2. Insert setup and process β. Go to step 8.

8. Reset $\{H_c, L_c, C_c\}$ to the index coordinate of the current job on the machine. Go to step 1.

Step 1 determines whether this decision process should be repeated or not, based on whether or not there are jobs left in the queue. If there are, then it sets the initial values. In step 2, job *i* is selected from the set R, and R is updated by removing job *i* from R. Then, the corresponding number of jobs in its setup-free zone, N_i is counted. Step 3 finds which jobs are located inside the setup-free zone of the current job on the machine and which are not (Figure 4.15). In step 4, if job *i* is inside the setup-free zone of the current job on the machine, N_i is compared to the current maximum value of N_α which is updated among jobs located in the setup-free zone of the current job on the machine. In step 5, if job *i* is outside the setup-free zone of the current job on the machine, N_i is compared to the current maximum value of N_β which is updated among jobs located outside the setup-free zone of the current job on the machine. Step 6 checks whether all the jobs in the queue have been considered or not. In step 7, if α exists, α is processed. If there is no α, setup is inserted and β is processed. Step 8 updates the index coordinate of the current job.

5.2.2. Performance Evaluation

According to Taner [2001], developing a lower bound for the problem of scheduling with sequence dependent setup times is equivalent to establishing a lower bound for the asymmetric traveling salesman problem with time windows. Therefore, developing a tight lower bound, which can be computed in a reasonable amount of time, is extremely difficult. Consequently, the performance of the proposed scheduling algorithm will be compared to that of an existing heuristic. Following Taner [2001]'s research, the modified LAO (β) rule is chosen as a benchmark. The modified scheduling algorithm with SST_D, Algorithm 1, and the modified LAO (β) rule, Algorithm 2, were implemented in the VF, and a variety of flowshop

scenarios were generated and solved with each algorithm. Then, the L_{max} values obtained with the two schemes were compared in order to evaluate their relative performance.

5.3. Experimental Framework

As previously mentioned, a flowshop setting in which only one of the machines have sequence dependent setups was generated. All jobs were processed on this machine in the middle of their routes. The number of operations was set equal to 3 since textile wet processing consists of a fabric preparation, a dyeing and a finishing operation. Problems in which there was only 1 machine of each type were considered (i.e., no machines in parallel).

5.3.1. Processing and Setup Time Estimation

The processing times for preparation and finishing, and the processing and setup times for dyeing were estimated from the dyeing literature.

5.3.1.1. Dyeing Machine

The processing times for dyeing operations were estimated from the data in Table 5.2 which is based on the work of Kulkarni et al. [1986]. Their data may not be representative of all operations since different dye manufacturing companies provide their own recommendations for dyeing profiles. Different fabric/dye types require different procedures, and each dyeing machine may need some adjustments. However, since these customizations do not change the processing time much on average, Kulkarni's data is still valid for estimation purposes.

Table 5.2. Processing times of each fabric type and corresponding dye type

Fabric type	Corresponding major dye group	Processing hours	Dyeing profile
Polyester	Disperse	3	Figure 5.2
Cotton	Direct dye	3.25	Appendix B. Figure B.1
Cotton	Reactive	3.5	Appendix B. Figure B.2
Wool	Acid dye	3	Appendix B. Figure B.3
Rayon	Direct dye	2.75	Appendix B. Figure B.4
Acrylic	Basic	3.25	Appendix B. Figure B.5
Polyamide	Acid dye	3.5	Appendix B. Figure B.6
Polyester/cotton	Disperse/reactive	7	Appendix B. Figure B.7

Since the experimentation will be for a given fabric/dye type, polyester fabric and disperse dye were selected. Polyester was chosen because it is widely used for various purposes, such as woven and knitted apparel. While other fabric types can be dyed with various dye groups, polyester fabric is usually dyed with the disperse dye group. In addition, the dyeing profile and corresponding preparation and finishing operations are simpler than that of other fabric/dye types.

Figure 5.2 shows the dyeing profile in which polyester is dyed using disperse dyes at 210°F in atmospheric exhaust with a sample beck machine at a 30:1 liquor ratio. However, since it is common to use a high-temperature dyeing machine for polyester dyeing, the dyeing profile was modified to raise the temperature to 266°F (130°C). It takes about 50 minutes to raise the temperature to 266°F at the rate of 2.3°F per minute for even dye absorption. Dyeing continues at this temperature for 90 minutes but this time can vary depending on the depth of the shade, circulation system etc. It takes about 35 minutes to cool down the temperature. After the dye bath is drained, fabric is rinsed and reduction-cleared for 10 minutes. Thus the processing time was estimated to be 3-4 hours (i.e., 50 + 90 + 35 + 10 + extra minutes).

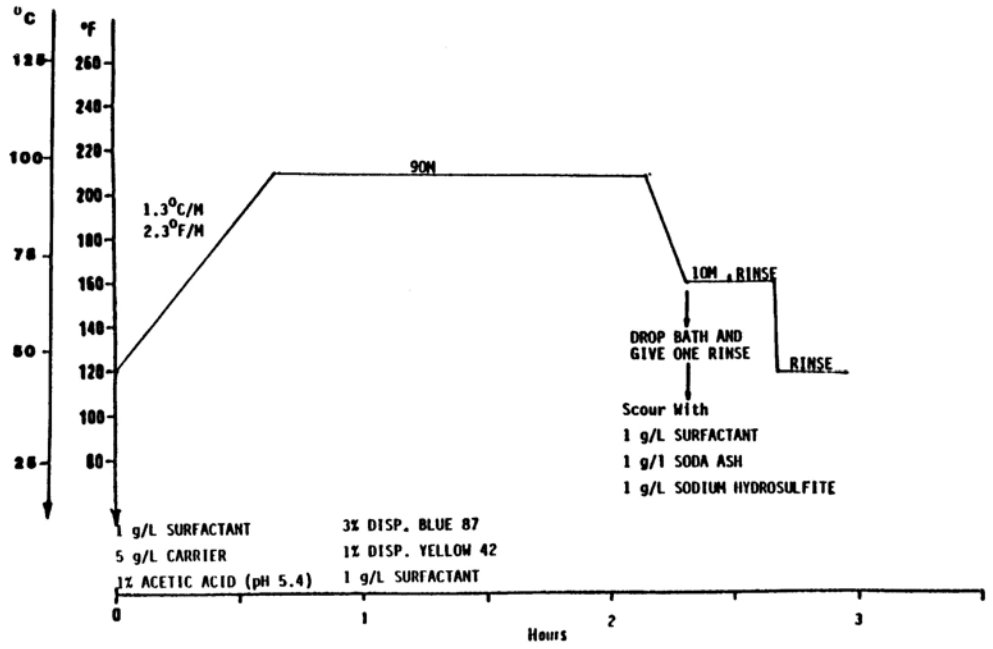


Figure 5.2. Dyeing polyester with disperse dyes [Kulkarni et al.,1986]

The setup time, in general, is equal to the time needed to boil out the dyeing machine. In the setup process, the temperature is raised to the maximum value of the dyeing profile, kept at that temperature, and then cooled down. To estimate the setup time, commercial cleaning agents for dyeing machines, NEOPOLAN KC-10 [Internet source (a), 2003] and Kayaclean F [Internet source (b), 2003] were investigated. For general cleaning of dyeing machines, NEOPLAN KC-10 should be added at a low temperature, raised to 130°C, kept at this temperature for 20-40 minutes, and then cooled down. Kayaclean F should be added at a low temperature, raised to 130°C, kept at this temperature for 30 minutes, and then cooled down. Based on this information, it was assumed that the temperature would be raised to 130°C, kept at that temperature for 30 minutes, then cooled down. Unlike the dyeing operation, the temperature can be raised without considering the dye absorption rate. Thus, it was assumed that it takes 30-50 minutes to both raise the temperature to 130°C and cool it

down. Therefore, the setup time was estimated as 110 minutes which is the average between 90-130 (i.e., 30-50 + 30 + 30-50) minutes.

5.3.1.2. Preparation Machine

In order to estimate the processing times for the other operations, the preparation of polyester textiles before dyeing and the finishing process after dyeing were investigated [Pajgrt, O. and Reichstadter, B (1979), and ICI (1964)]. In general, scouring, bleaching and setting are common preparation operations before dyeing. Polyester is usually scoured to get rid of traces of oils, sizes and other processing aids, since their irregular distribution may cause variation in shades after dyeing. If unset fabrics are processed in rope form, creases may be set in and the time and temperature of the processing determines the extent of the creases. Fabrics that have been heat-set at temperatures of 180-200°C tend to crease less in wet processing and, moreover, any formed creases are more readily removed by re-stentering. The processing times of scouring and setting vary considerably with the type of goods, and detailed recommendations are summarized in Table 5.3.

Table 5.3. Preparation of polyester fabric for dyeing [ICI, 1964]

Type of goods	Scouring	Setting
Continuous-filament fabrics	Boil for 15-30 min. on jig 60°C for 15-30 min. on winch	Stenter-set at 180-220°C
Staple fabric from short-staple yarns	60-75°C for 15-30 min.	Stenter-set at 180-200°C
Staple fabric from long-staple yarns	40°C for 10 min Clean for 30-45 min.	Stenter-set at 180-200°C
Stretch polyester	60°C for 15-30 min.	In steam for 2 min. Infra-red setter at 180°C for 1.5 min.

For 100% polyester textiles, no bleaching treatment is necessary before dyeing since they are generally sufficiently white. However, for brilliant pastel shades, polyester fabrics need to be bleached before dyeing to increase whiteness. An oxidation bleaching with 80% sodium chlorite powder is often used. It takes 20 to 30 minutes for the bath to boil, and the bleaching continues to boil for 30 to 45 minutes. In addition to these preparations, according to Pajgrt and Reichstadter [1979], woven fabric with high polyester fiber content tends to have a somewhat stiff and harsh hand so that a softening treatment is required. A strong alkaline treatment (i.e., caustic soda) is often applied before dyeing because it causes discoloration.

Based on the available information, the processing time of the preparation operation before dyeing was estimated as 100-190 minutes. The scouring process takes 90 minutes since the temperature is raised for 30 minutes, kept at that temperature for 30 minutes on average, and cooled down for 30 minutes. The setting operation was estimated to take 10 minutes. Finally, bleaching and softening might be added based on the situation. This extra 90 minutes was also considered in the processing time estimation.

5.3.1.3. Finishing Machine

For the finishing process, only an antistatic finish was considered since most of the finishing procedures noted were for polyester blend fabrics [Pajgrt, O. and Reichstadter, B (1979), and ICI (1964)]. In order to avoid a tendency to stick to the body and the generation of static electricity during subsequent processing, antistatic agents are often used in the processing of 100 % polyester fabrics. Besides an antistatic agent, a lubricant may be applied,

depending on the subsequent processing operations or on the end-use. Processing time data is given in Table 5.4.

Table 5.4. Processing time data of some commercial antistatic agents
[Pajgrt, O. and Reichstadter, B, 1979]

Trade mark	Producer	Processing time
Arkostat P	Hoechst	Drying at 80°C and 1 min. at 140°C
Spolapret CS	Chemapol	Drying at 80°C and 10 min. at 160°C
Cerol WB	Sandoz	Drying at 80°C and 15 min. at 110°C
Solacrol 20	Hungary	Drying at 80°C and 5 min. at 140°C
Syntegal TE	Chemapol	Drying at 80°C and 5 min. at 140°C

The processing time of the finishing operation was estimated to be 65-75 minutes. The temperature is raised for 30 minutes, remains for 5-15 minutes on average, and is cooled down for 30 minutes.

5.3.2. Index Coordinates Generation

Before generating index coordinates for the jobs, we must address the concern of whether the true colors can be realized by dyeing with the given fabric and colorants. To accomplish this, the SLI-Form developed by Gretag Macbeth (version 2.80) was utilized, which is a commercial program for color formulation and quality control. The program allows the user to select a target color, a set of colorants, and a substrate in order to find a recipe for reproducing that target with the specified colorants. For 100% polyester fabric, the commercial dye group of Dystar DIANIX Blue AC-E, RED AC-E, YELLOW AC-E, and WHITE BYER 10GN200 was selected from the database. However, the results of this formulation experimentation revealed that this combination of dye group could generate only a very limited color region. Even though other combinations of dye groups from the database

may generate the remaining color region, it would be difficult to check all those possible combination of dye groups. Moreover, there might be other dye groups which are not stored in the database that was used. Therefore, in this dissertation, index coordinates will be generated randomly with the assumptions that the cylindrical color space represents the true color space and those true colors can be realized with some combination of dye groups.

5.3.3. Experimental Design

The flowshop generated is illustrated in Figure 5.3. The processing times for the dyeing machine are sampled from a Uniform distribution with a range between 600 and 800 time units. One hour is equivalent to 200 time units in this conversion. The setup time for the dyeing machine is set to 366 time units (Refer to 5.3.1.1). The processing times for fabric preparation are sampled from a Uniform distribution with a range between 333 and 633 time units. The processing times for the finishing operation are sampled from a Uniform distribution with a range between 216 and 250 time units (Refer to 5.3.1.2).

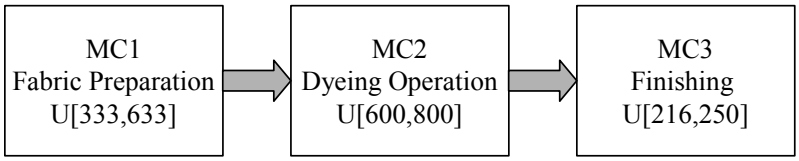


Figure 5.3. Flow shop setting

The hue index of each job is sampled from a Uniform distribution with a range between 1 and 8. The lightness index of each job is sampled from a Uniform distribution with a range between 1 and 10. The chroma index of each job is sampled from a Uniform

distribution with a range between 1 and 5.

The due dates are sampled from a Uniform distribution with a range between 0 and $\bar{d}$ -1, where $\bar{d}$ is the due date range parameter. $\bar{d}$ was varied based on Ω and it was defined in the following two different ways:

- Method 1: $\Omega = (O-1)\times \bar{P} + \bar{p} + 0.5s$
- Method 2: $\Omega = (\bar{p} + 0.5s) \times$Number of Jobs

where $\bar{P}$ and $\bar{p}$ are the average processing time of regular machines and that of sequence dependent machine, respectively. s is the setup time on the machine with sequence dependencies. Following Taner [2001]'s generation method, Method 1, the due date is proportional to the average total processing time of a job. Since this method does not consider the number of jobs in the factory, this seemed to generate extremely tight due dates and meeting them was not usually possible for all jobs. Also recall that the processing time of the dyeing operation is greater than each of the other operations. Thus, a more realistic due date range is closely related to the time needed in order to finish the dyeing operation of the last job in the flowshop. Therefore, a new generation method, Method 2, was developed to test higher due date ranges in relation to the average processing time of the dyeing operation and the number of jobs in the factory. For each due date generation method, due date ranges are generated by setting $\bar{d}$ equal to 0.5Ω, 0.75Ω, Ω, 1.25Ω and 2Ω. Scenarios with 20, 50 and 100 jobs are generated. 20 random instances (replications) are generated for each parameter combination resulting in a total of 300 problems.

Experimentation was undertaken to determine the best values of the parameters, T, β, γ, and τ. As a base case, T was set to 3200, β to 3, γ to 0 and τ to 3200. When one parameter

was varied, the others were set to the values of the base case. Whenever a set of parameters, T, β, γ, and τ, was changed, 300 problems (i.e., 5 range multipliers × 3 number of jobs × 20 replications) under due date generation Method 1 and 300 problems under due date generation Method 2 were generated. T was varied to be 0, 1600, 3200, and 4800. The values of 3, 4, and 5 were tested to determine β. As β increased, the average performance became better but the computational time increased. Since the improvement in performance was small considering the increase in computational time and previous research [Taner (2001), Ovacik and Uzsoy (1994a and 1994b)] used the value of 3 for β, $\beta = 3$ was selected for further experimentation. γ and τ each were varied to be 0, 1600, 3200, 4800, and this generated 16 combinations of γ and τ. After determining the parameter set of γ and τ which performed best (i.e., 0 and 4800, respectively), these two parameters were refined by varying γ and τ on a smaller scale. Based on the results of these initial experimentations, T was set to 3200, β to 3, γ to 400, and τ to 4800 in further experimentation.

5.4. Experimental Results with Algorithm 1

To compare the performance of Algorithm 1 and Algorithm 2, the average of the difference calculated by subtracting the L_{max} of Algorithm 1 from the L_{max} of Algorithm 2 was computed. Positive values represent that Algorithm 1 performs better than Algorithm 2. To better enable direct comparison of the performance in different due date ranges, the ratio of the average difference of L_{max} to the maximum due date, d_{max}, was also calculated.

5.4.1. Initial Results

Table 5.5 and Table 5.6 show the overall performance and computational time of Algorithm 1 and 2 under due date generation Method 1 and 2, respectively. Problems were generated varying the range multiplier and number of jobs.

Table 5.5. Comparison of performance and computational time of Algorithm 1 and 2 under due date generation Method 1

Range Multiplier	Number of Jobs	L_{max} Algorithm 1	L_{max} Algorithm 2	d_{max}	L_{max} Difference, L_{max} (Algorithm2) - L_{max} (Algorithm1)	L_{max} Difference/ d_{max} (%)	Computational Time Algorithm 1	Computational Time Algorithm 2
0.5	20	15556.95	16278.00	753.15	721.05	95.74	0.055	0.061
0.5	50	37160.90	40350.20	781.25	3189.30	408.23	0.286	0.166
0.5	100	72474.75	81639.50	791.00	9164.75	1158.63	1.296	0.364
0.75	20	15168.20	15979.75	1130.60	811.55	71.78	0.053	0.061
0.75	50	36887.05	39992.85	1172.55	3105.80	264.88	0.277	0.176
0.75	100	72193.65	80845.45	1187.35	8651.80	728.66	1.250	0.353
1	20	14809.15	15585.90	1507.75	776.75	51.52	0.051	0.063
1	50	36546.60	39744.00	1563.80	3197.40	204.46	0.349	0.162
1	100	71714.25	80569.00	1583.55	8854.75	559.17	1.138	0.371
1.25	20	14516.25	15260.00	1884.05	743.75	39.48	0.054	0.060
1.25	50	36118.95	39305.30	1954.20	3186.35	163.05	0.270	0.185
1.25	100	71402.40	80160.15	1978.85	8757.75	442.57	1.346	0.398
2	20	13586.60	14157.35	3016.10	570.75	18.92	0.053	0.061
2	50	35221.70	38254.60	3128.10	3032.90	96.96	0.348	0.162
2	100	70510.50	78840.70	3167.60	8330.20	262.98	1.190	0.363

Due dates generated by Method 1 tend to be smaller than those generated by Method 2. With tighter due dates (i.e., Method 1), Algorithm 1 performed much better than Algorithm 2. With respect to the average difference in L_{max}, Algorithm 1 performed much better on problems with a large number of jobs but only somewhat better on those with a small number of jobs. The ratio of the difference of L_{max} to the maximum due date varied from 18.92 % to 1158 %.

Table 5.6. Comparison of performance and computational time of Algorithm 1 and 2 under due date generation Method 2

Range Multiplier	Number of Jobs	L_{max} Algorithm 1	L_{max} Algorithm 2	d_{max}	L_{max} Difference, L_{max} (Algorithm2) - L_{max} (Algorithm1)	L_{max} Difference/ d_{max} (%)	Computational Time Algorithm 1	Computational Time Algorithm 2
0.5	20	10181.20	9254.10	8328.85	-927.10	-11.13	0.051	0.053
0.5	50	23190.75	20865.80	21595.85	-2324.95	-10.77	0.248	0.151
0.5	100	48773.10	40746.45	43737.30	-8026.65	-18.35	1.175	0.311
0.75	20	7049.35	5547.60	12493.65	-1501.75	-12.02	0.052	0.055
0.75	50	14742.20	11236.35	32393.40	-3505.85	-10.82	0.292	0.151
0.75	100	26418.05	20926.60	65606.30	-5491.45	-8.37	1.343	0.313
1	20	3523.75	2838.60	16658.25	-685.15	-4.11	0.055	0.052
1	50	4278.75	3524.80	43192.15	-753.95	-1.75	0.382	0.145
1	100	5924.85	5168.90	87475.15	-755.95	-0.86	1.319	0.328
1.25	20	1980.70	1727.60	20822.90	-253.10	-1.22	0.037	0.039
1.25	50	1536.75	1251.75	53989.75	-285.00	-0.53	0.230	0.113
1.25	100	1929.65	1622.65	109344.15	-307.00	-0.28	1.174	0.246
2	20	388.30	293.35	33317.00	-94.95	-0.28	0.020	0.024
2	50	162.40	63.90	86384.70	-98.50	-0.11	0.157	0.065
2	100	287.90	288.25	174950.80	0.35	0.00	0.506	0.130

For all the problems generated by Method 2 except those with range multiplier 2 and 100 jobs, Algorithm 2 performed better on average than Algorithm 1. The difference in L_{max} as a percentage of the maximum due date varied from -18.35 % to -0.11% for the problems in which Algorithm 2 performed better. When the number of jobs was increased and the range multiplier was held constant, the absolute value of the difference in L_{max} also increased except with a range multiplier of 2 and 100 jobs. When the range multiplier was increased and the number of jobs was held constant, the absolute value of the ratio tended to decrease except with a range multiplier of 0.75.

In general, when scheduling with sequence dependent setup times in low due date ranges, minimizing setups tends to reduce maximum lateness. When the due date range is high, scheduling jobs in order of earliest due date tends to minimize maximum lateness.

Since Algorithm 1 tends to choose the SST_D rule (i.e. *SUBSET job* or *SST job*) more often in the low due date ranges (i.e., all due date ranges under Method 1), it often schedules jobs to minimize setup times. Thus, Algorithm 1 performs significantly better than Algorithm 2 with tight due dates generated by Method 1. On the other hand, while Algorithm 1 still frequently selects the SST_D rule in the higher due date ranges (i.e., all due date ranges under Method 2), Algorithm 2 seems to have a better balance between minimizing setups and meeting due dates. Consequently, the performance of Algorithm 1 may be improved on problems generated by Methods 2 if the condition in which Algorithm 1 selects the *EDD job* is refined.

The average computational times on a 1.5 GHz PC were slightly lower for Algorithm 1 than Algorithm 2 for small numbers of jobs. As the number of jobs increased, the computational time for Algorithm 1 increased because time for the job-counting procedure increased by a second order polynomial function. If there are n jobs in the queue and job i is the candidate for the next job, then there are n-1 jobs in the queue to check whether they require a setup if they follow job i. Thus, the computational time is proportional to $n\times(n\text{-}1)$.

5.4.2. Algorithm 1 Modifications

In order to improve the performance of Algorithm 1 on problems generated by Method 2, the following modifications were considered:

M1. Higher Priority of *EDD job* over *SUBSET job*

In Algorithm 1, if all jobs in the queue are already tardy and at least one is tardy by γ or more time units, the SUBSET rule is used. This rule picks a job which

may save setup time instead of the most critical job which may have the maximum lateness. Step 4.1 in Algorithm 1 was modified as below:

Step 4.1. If all the jobs waiting in the queue are already tardy and at least one is tardy by γ or more time units, after finding both the jobs selected by the EDD rule and the SUBSET rule, determine whether the *EDD job* or *SUBSET job* should be chosen as follows:

- If the *EDD job* is more urgent than the *SUBSET job* by the initial value of τ or more, choose the *EDD job*.
- Otherwise, choose the *SUBSET job*.

The initial τ value represents the lower limit on the time remaining (*Revised Slack* minus the maximum of *Available Time* and *Present Time*) for a job to be considered not urgent (Figure 5.4). Thus, the initial τ value was used as the lower limit of the time difference between the *EDD job* and the *SUBSET job* (Figure 5.5).

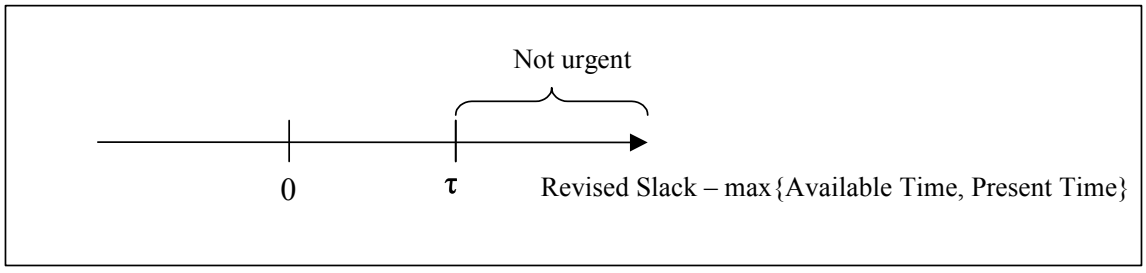


Figure 5.4. Graphic illustration of the meaning of initial τ

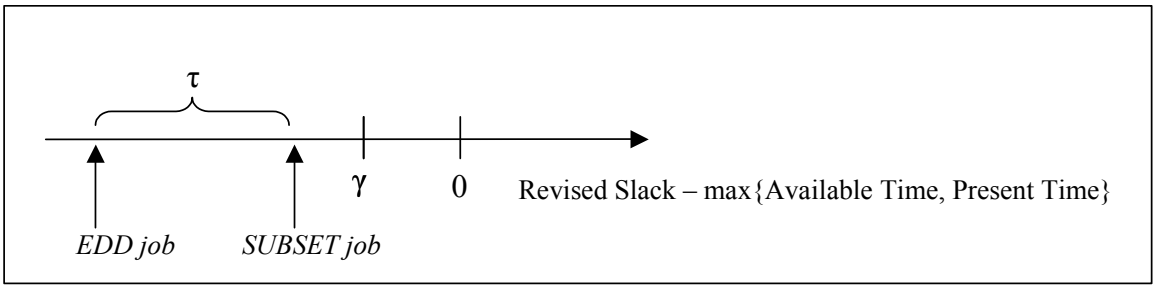


Figure 5.5. Graphic illustration of the condition added into step 4.1

M2. Higher Priority of *EDD job* over *SST job*

Algorithm 1 chooses the *SST job* if the condition that the *SST job* has more than τ time units fails, even if there are extremely tardy jobs and the *SST job* is not tardy (Figure 5.6). Thus step 4.2 in Algorithm 1 was modified as below:

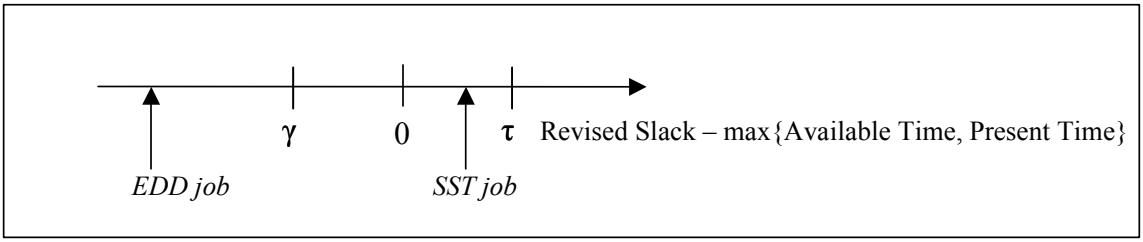


Figure 5.6. Graphic illustration of a bad decision by Algorithm 1

Step 4.2. If the *SUBSET* rule is not chosen, find both the jobs selected by the EDD rule and STT_D rule and determine whether the *EDD job* or *SST job* should be chosen as follows:

- If the *SST job* has τ time units or more until its due date while the *EDD job* is more urgent, choose the *EDD job*.

- If the *SST job* has less than τ time units but it is not tardy while the *EDD job* is extremely tardy by γ time units or more, choose the *EDD job*.
- Otherwise, choose the *SST job*.

M3. Varying C*ounter*, the number of jobs in the consideration list

In Algorithm 1, all jobs in the queue were added to the consideration list. In order to reduce the computational time, the *Counter* was varied and problems were generated for each *Counter* in the same manner as in section 5.4.1 except that due dates were generated only by Method 2. Figure 5.7 and 5.8 display the average $L_{\max}$ and the average computational time, respectively, for a range of *Counter* values. Setting the *Counter* to 4 resulted in the best performance, and $L_{\max}$ increased as the *Counter* was increased above 4. This is an interesting result in that, intuitively, using a larger *Counter* would lead to better performance since more possibilities would be considered. With a smaller *Counter*, the *SST job* or the *SUBSET job* is chosen among jobs which have smaller revised slack than when considering all the jobs in the queue. If the *SST job* or the *SUBSET job* is selected as the next job when using a smaller *Counter*, then this decision is based on a combination of the EDD rule and the SST_D rule rather than on the SST_D rule alone. This may be the reason for the better performance. As expected, the average computational time increased as the *Counter* increased. Recall that Algorithm 2 also considers 3 jobs in the queue as candidates for the next job. In Figure 5.7, a *Counter* of 3 and 4 showed similar performance. Thus, even though using a *Counter* of 4 gave the best performance, Algorithm 1 was modified to consider 3 jobs in the queue instead of all jobs.

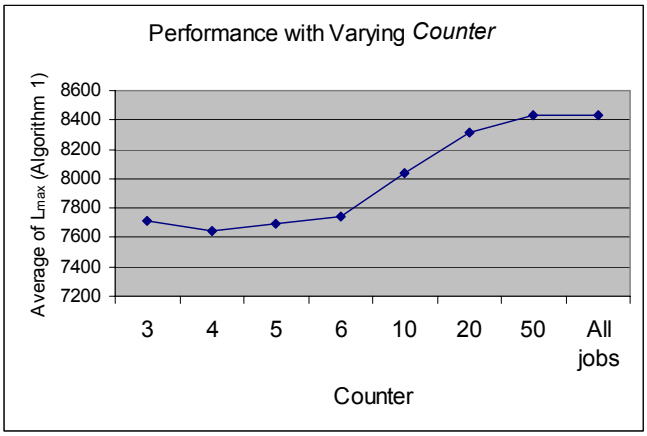


Figure 5.7. The average $L_{\max}$ with varying *Counter* values

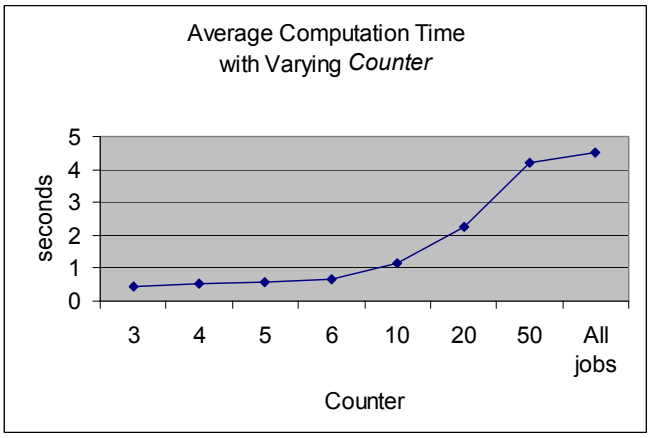


Figure 5.8. The average computational time with varying *Counter* values

M4. Combination of M1 and M2

M5. Combination of M1, M2 and M3

5.4.3. Performance of Algorithm 1 Modifications on Problems Generated by Method 2

Table 5.7 shows the original L_{max} value of Algorithm 1 and the difference between the original value and the L_{max} value after modifications when due date generation Method 2 was used. Positive values represent that the modification improves the performance of Algorithm 1. M1 improved the performance in the low due date ranges (i.e., range multiplier of 0.5 and 0.75). M2 improved the performance with small number of jobs, and in the middle and high due date ranges (i.e., range multiplier of 1, 1.25 and 2). M3 improved the performance in the low and middle due date ranges (i.e., range multiplier of 0.5, 0.75, and 1). M4 and M5 resulted in performance improvement in all due date ranges. For most types of problems, M5 performed better than M4. Therefore, M5 was selected as the best algorithm to be used in subsequent experiments

Table 5.7. The results of Algorithm 1 modifications

Range Multiplier	Number of Jobs	L_{max} (Algorithm 1)	L_{max} (Algorithm 1) - L_{max} (Modification)				
			M1	M2	M3	M4	M5
0.5	20	10181.20	203.30	56.45	991.05	227.00	991.05
0.5	50	23190.75	3028.40	-135.35	3560.10	3068.10	3541.80
0.5	100	48773.10	10216.15	-522.40	11693.30	10229.45	11826.50
0.75	20	7049.35	39.20	759.15	1090.00	759.15	984.40
0.75	50	14742.20	2441.20	1211.20	4818.40	2829.95	4685.00
0.75	100	26418.05	6275.75	-1007.60	9299.15	6712.30	9014.05
1	20	3523.75	40.95	235.25	218.10	235.25	367.05
1	50	4278.75	0.00	654.55	496.55	654.55	703.00
1	100	5924.85	0.00	339.50	901.45	339.50	1128.05
1.25	20	1980.70	0.00	148.95	29.35	148.95	157.75
1.25	50	1536.75	0.00	110.05	-105.20	110.05	177.10
1.25	100	1929.65	0.00	233.65	-99.55	233.65	292.40
2	20	388.30	0.00	65.05	-18.35	65.05	46.70
2	50	162.40	0.00	56.10	0.00	56.10	43.90
2	100	287.90	0.00	7.05	0.00	7.05	7.05

Table 5.8 shows the overall performance and computational time comparison of modification M5 and Algorithm 2 on problems generated by Method 2. Even though M5 significantly improved the performance of Algorithm 1 in all due date ranges, there are still several problem types where M5 performs worse than Algorithm 2 on average. With a large numbers of jobs or in the low due date ranges (i.e., range multiplier of 0.5), Algorithm 1 tended to perform better than Algorithm 2. However, with a small number of jobs or in the high due date ranges, Algorithm 2 performed somewhat better than Algorithm 1. Average computational time significantly decreased compared to Algorithm 1, especially with 100 jobs. However, it is not less than that of Algorithm 2 for all types of problems.

Table 5.8. Comparison of performance and computational time of Algorithm 2 and M5 with due dates generated by Method 2

Range Multiplier	Number of Jobs	L_{max}		d_{max}	L_{max} Difference, L_{max} (Algorithm2) - L_{max} (M5)	L_{max} Difference/ d_{max} (%)	Computational Time	
		M5	Algorithm 2				M5	Algorithm 2
0.5	20	9190.15	9254.10	8328.85	63.95	0.77	0.047	0.053
0.5	50	19648.95	20865.80	21595.85	1216.85	5.63	0.145	0.151
0.5	100	36946.60	40746.45	43737.30	3799.85	8.69	0.344	0.311
0.75	20	5983.80	5547.60	12493.65	-517.35	-4.14	0.048	0.055
0.75	50	10057.20	11236.35	32393.40	1179.15	3.64	0.153	0.151
0.75	100	17404.00	20926.60	65606.30	3522.60	5.37	0.356	0.313
1	20	3156.70	2838.60	16658.25	-318.10	-1.91	0.045	0.052
1	50	3575.75	3524.80	43192.15	-50.95	-0.12	0.170	0.145
1	100	4784.35	5168.90	87475.15	372.10	0.43	0.338	0.328
1.25	20	1851.05	1727.60	20822.90	-95.35	-0.46	0.039	0.039
1.25	50	1388.55	1251.75	53989.75	-107.90	-0.20	0.127	0.113
1.25	100	1645.70	1622.65	109344.15	-14.60	-0.01	0.281	0.246
2	20	341.60	293.35	33317.00	-48.25	-0.14	0.020	0.024
2	50	106.30	63.90	86384.70	-54.60	-0.06	0.074	0.065
2	100	280.25	288.25	174950.80	7.40	0.00	0.141	0.130

5.5. Experimental Results with Policy 2

5.5.1. Initial Results

Recall that Algorithm 1 is a modification of Taner [2001]'s scheduling algorithm, Policy 2 (See Section 2.3.2.1). The cause of the worse performance, in comparison to Algorithm 2, of Algorithm 1 and even that of M5 with a small number of jobs and in the high due date ranges might be from either the original algorithm, Policy 2, or the modifications including the new SST_D rule. Thus, the same 3 machine flowshop problem was scheduled using Policy 2 with Taner's setup matrix (Table 2.5). Recall that instead of the index coordinates, every job has a family index from 1 to 6. T was set to 150, β to 3, γ to 0, and τ to 150 following Taner's original experimental parameters.

Table 5.9 and Table 5.10 show the overall performance and computational time of Policy 2 and Algorithm 2 under due date generation Method 1 and 2, respectively.

Table 5.9. Comparison of performance and computational time of Policy 2 and Algorithm 2 under due date generation Method 1 with $T = 150$, $\beta = 3$, $\gamma = 0$, $\tau = 150$

Range Multiplier	Number of Jobs	L_{max}		d_{max}	L_{max} Difference, L_{max} (Algorithm 2) - L_{max} (Policy 2)	L_{max} Difference/ d_{max} (%)	Computational Time	
		Policy 2	Algorithm 2				Policy 2	Algorithm 2
0.5	20	14857.15	15451.05	753.15	593.90	78.86	0.020	0.026
0.5	50	36306.05	38999.15	781.25	2693.10	344.72	0.060	0.072
0.5	100	71650.15	77538.30	791.00	5888.15	744.39	0.153	0.133
0.75	20	14509.35	15105.55	1130.60	596.20	52.73	0.020	0.026
0.75	50	35923.25	38650.85	1172.55	2727.60	232.62	0.060	0.072
0.75	100	71255.15	77296.05	1187.35	6040.90	508.77	0.153	0.133
1	20	14138.25	14774.25	1507.75	636.00	42.18	0.020	0.024
1	50	35511.05	38129.60	1563.80	2618.55	167.45	0.074	0.061
1	100	70885.35	76821.60	1583.55	5936.25	374.87	0.156	0.131
1.25	20	13767.65	14374.55	1884.05	606.90	32.21	0.020	0.023
1.25	50	35089.50	37744.10	1954.20	2654.60	135.84	0.061	0.062
1.25	100	70485.30	76467.75	1978.85	5982.45	302.32	0.150	0.129
2	20	12839.30	13347.20	3016.10	507.90	16.84	0.019	0.024
2	50	33986.90	36481.45	3128.10	2494.55	79.75	0.074	0.062
2	100	69369.85	75288.30	3167.60	5918.45	186.84	0.164	0.127

Table 5.10. Comparison of performance and computational time of Policy 2 and Algorithm 2 under due date generation Method 2 with $T = 150$, $\beta = 3$, $\gamma = 0$, $\tau = 150$

Range Multiplier	Number of Jobs	L_{max}		d_{max}	L_{max} Difference, L_{max} (Algorithm 2) - L_{max} (Policy 2)	L_{max} Difference/ d_{max} (%)	Computational Time	
		Policy 2	Algorithm 2				Policy 2	Algorithm 2
0.5	20	8573.35	8471.80	8328.85	-101.55	-1.22	0.019	0.025
0.5	50	19465.05	19361.45	21595.85	-103.60	-0.48	0.063	0.061
0.5	100	36081.00	37482.55	43737.30	1401.55	3.20	0.163	0.125
0.75	20	5755.10	4875.85	12493.65	-879.25	-7.04	0.019	0.025
0.75	50	11924.00	9635.80	32393.40	-2288.20	-7.06	0.063	0.061
0.75	100	20886.85	17716.70	65606.30	-3170.15	-4.83	0.163	0.125
1	20	2724.40	2490.35	16658.25	-234.05	-1.41	0.018	0.021
1	50	3560.55	2901.20	43192.15	-659.35	-1.53	0.061	0.058
1	100	4862.95	4163.50	87475.15	-699.45	-0.80	0.147	0.125
1.25	20	1558.55	1490.20	20822.90	-68.35	-0.33	0.014	0.016
1.25	50	1394.10	1198.75	53989.75	-195.35	-0.36	0.043	0.041
1.25	100	1662.00	1547.80	109344.15	-114.20	-0.10	0.127	0.095
2	20	251.35	251.35	33317.00	0.00	0.00	0.010	0.009
2	50	76.15	64.05	86384.70	-12.10	-0.01	0.035	0.031
2	100	322.20	257.00	174950.80	-65.20	-0.04	0.075	0.058

Similar to the results in section 5.4.1, with extremely tight due dates (i.e., Method 1) Policy 2 performs significantly better than Algorithm 2. The ratio of the difference of L_{max} to the maximum due date varies from 16.84 % to 744.39 %. In the problems generated by Method 2, Algorithm 2 performs somewhat better on average than Policy 2. The ratio of the difference of L_{max} to the maximum due date varies from -7.04 % to -0.01%.

To allow more direct comparison to the results in section 5.4.1, T was set to 3200, β to 3, γ to 400, and τ to 4800 and the same experimentation was repeated. Table 5.11 and Table 5.12 show the overall performance and computational time of Policy 2 and Algorithm 2 under due date generation Method 1 and 2, respectively. The results showed a similar pattern of results to Table 5.9 and 5.10. Based on the fact that the trend of performance of Policy 2 was the same as that of Algorithm 1, it can be concluded that much of the worse performance of Algorithm 1 on problems generated by Method 2 probably came from the original algorithm, Policy 2.

Table 5.11. Comparison of performance and computational time of Policy 2 and Algorithm 2 under due date generation Method 1 with $T = 3200$, $\beta = 3$, $\gamma = 400$, $\tau = 4800$

Range Multiplier	Number of Jobs	L_{max}		d_{max}	L_{max} Difference, L_{max} (Algorithm 2) - L_{max} (Policy 2)	L_{max} Difference/ d_{max} (%)	Computational Time	
		Policy 2	Algorithm 2				Policy 2	Algorithm 2
0.5	20	14912.75	15470.55	753.15	557.80	74.06	0.020	0.028
0.5	50	36332.75	39096.20	781.25	2763.45	353.72	0.056	0.073
0.5	100	71665.70	77593.20	791.00	5927.50	749.37	0.152	0.151
0.75	20	14536.15	15131.30	1130.60	595.15	52.64	0.020	0.026
0.75	50	35957.45	38690.00	1172.55	2732.55	233.04	0.060	0.072
0.75	100	71274.10	77307.25	1187.35	6033.15	508.12	0.153	0.133
1	20	14196.65	14808.75	1507.75	612.10	40.60	0.020	0.024
1	50	35529.05	38111.30	1563.80	2582.25	165.13	0.074	0.061
1	100	70908.05	76853.90	1583.55	5945.85	375.48	0.156	0.131
1.25	20	13830.60	14392.80	1884.05	562.20	29.84	0.020	0.023
1.25	50	35112.75	37725.80	1954.20	2613.05	133.71	0.061	0.062
1.25	100	70504.30	76513.00	1978.85	6008.70	303.65	0.150	0.129
2	20	12888.35	13293.75	3016.10	405.40	13.44	0.019	0.024
2	50	33998.60	36451.75	3128.10	2453.15	78.42	0.074	0.062
2	100	69423.40	75275.05	3167.60	5851.65	184.73	0.164	0.127

Table 5.12. Comparison of performance and computational time of Policy 2 and Algorithm 2 under due date generation Method 2 with $T = 3200$, $\beta = 3$, $\gamma = 400$, $\tau = 4800$

Range Multiplier	Number of Jobs	L_{max}		d_{max}	L_{max} Difference, L_{max} (Algorithm 2) - L_{max} (Policy 2)	L_{max} Difference/ d_{max} (%)	Computational Time	
		Policy 2	Algorithm 2				Policy 2	Algorithm 2
0.5	20	8764.65	8454.95	8328.85	-309.70	-3.72	0.020	0.023
0.5	50	19543.50	19329.00	21595.85	-214.50	-0.99	0.061	0.062
0.5	100	36338.25	37422.90	43737.30	1084.65	2.48	0.168	0.130
0.75	20	5691.20	4875.85	12493.65	-815.35	-6.53	0.019	0.025
0.75	50	11410.80	9598.65	32393.40	-1812.15	-5.59	0.063	0.061
0.75	100	20651.65	17698.25	65606.30	-2953.40	-4.50	0.163	0.125
1	20	2924.25	2481.95	16658.25	-442.30	-2.66	0.018	0.021
1	50	3437.05	2882.35	43192.15	-554.70	-1.28	0.061	0.058
1	100	4979.60	4163.35	87475.15	-816.25	-0.93	0.147	0.125
1.25	20	1667.45	1490.20	20822.90	-177.25	-0.85	0.014	0.016
1.25	50	1378.75	1183.05	53989.75	-195.70	-0.36	0.043	0.041
1.25	100	1722.10	1547.65	109344.15	-174.45	-0.16	0.127	0.095
2	20	258.30	251.35	33317.00	-6.95	-0.02	0.010	0.009
2	50	89.15	51.10	86384.70	-38.05	-0.04	0.035	0.031
2	100	289.55	256.85	174950.80	-32.70	-0.02	0.075	0.058

5.5.2. Performance of Policy 2 Modifications on Problems Generated by Method 2

In order to improve the performance of Policy 2 on problems generated by Method 2, the following modifications were considered:

M1'. γ interpretation

There were inconsistencies in Taner's definition of γ. In Policy 2, Taner states that when the condition that at least one job was tardy by γ or more time units was satisfied, the SUBSET rule is used. Thus, Taner computed the lateness of a job by subtracting the revised slack of the job from the maximum of its available time and the present time. However, in another definition which Taner stated, γ is an upper

limit on the length of time that a job should have until its due date to be considered extremely tardy. This difference would be computed by subtracting the maximum of the job's available time and the present time from the revised slack of the job. Therefore, this computation was implemented in this modification.

M2'. "All tardy" condition modification

In Policy 2, the SUBSET rule is considered only when all jobs are tardy. Whether this condition is met or not closely relates to the due date range. For example, the chance of meeting this condition is higher in the low due date ranges than the high due date ranges. However, the SUBSET rule should be used in all cases, including the case in which there are several extremely tardy jobs but one that is not tardy. Thus, after eliminating this condition, when at least one extremely tardy job exists, the *SUBSET job* and the *EDD job* will be found and the next job will be chosen among these.

M3'. Combination of M1' and M2'

For each modification, T was set to 3200, β to 3, γ to 400, and τ to 4800 and the same experimentation was repeated. Table 5.13 shows the original L_{max} value of Policy 2, the L_{max} difference between the original value of Policy 2 and that after modifications, and the L_{max} difference between Algorithm 2 and Policy 2 after modifications.

Table 5.13. The results of Policy 2 Modifications

Range Multiplier	Number of Jobs	L_{max} (Policy 2)	L_{max} (Policy 2) - L_{max} (Modification)			L_{max} (Algorithm 2) - L_{max} (Modification)		
			M1'	M2'	M3'	M1'	M2'	M3'
0.5	20	8764.65	125.05	-105.75	1.70	-184.65	-415.45	-308.00
0.5	50	19543.50	18.15	-244.00	-240.80	-196.35	-458.50	-455.30
0.5	100	36338.25	-68.15	17.60	-147.40	1016.50	1102.25	937.25
0.75	20	5691.20	318.75	390.15	-6.90	-496.60	-425.20	-822.25
0.75	50	11410.80	382.30	1439.75	606.20	-1429.85	-372.40	-1205.95
0.75	100	20651.65	346.40	3591.55	4020.05	-2607.00	638.15	1066.65
1	20	2924.25	-46.45	107.00	55.00	-488.75	-335.30	-387.30
1	50	3437.05	37.20	338.20	455.95	-517.50	-216.50	-98.75
1	100	4979.60	21.75	1151.80	1022.55	-794.50	335.55	206.30
1.25	20	1667.45	-144.50	-75.40	-102.00	-321.75	-252.65	-279.25
1.25	50	1378.75	-188.10	-303.30	-204.30	-383.80	-499.00	-400.00
1.25	100	1722.10	-141.95	-209.35	-209.55	-316.40	-383.80	-384.00
2	20	258.30	-51.05	-180.35	-180.35	-58.00	-187.30	-187.30
2	50	89.15	-47.15	24.40	-20.05	-85.20	-13.65	-58.10
2	100	289.55	-47.10	-95.65	-105.65	-79.80	-128.35	-138.35

For interpreting the first L_{max} difference, positive values represent that the modification improves the performance of Policy 2. M1' and M2' sometimes improved the performance in the low and middle due date ranges (i.e., Range multiplier of 0.5, 0.75 and 1), but the performance in the high due date range deteriorated after the modifications. Combining M1' and M2' did not usually yield a greater improvement than both M1' and M2 alone. For interpreting the second L_{max} difference, positive values represent the modified Policy 2 performs better than Algorithm 2. Even though these modifications improved the performance of Policy 2 in some due date ranges, Policy 2 still does not perform better than Algorithm 2 in most due date ranges.

5.6. Experimental Results with Algorithm 3

5.6.1. Initial Results

Based on results of previous modifications, Algorithm 3 which is the combination of M1, M2, M3, M1' and M2' was tested on the same problems. T was set to 3200, β to 3, γ to 400, and τ to 4800. Table 5.14 and Table 5.15 show the overall performance and computational time of Algorithm 3 under Method 1 and Method 2, respectively.

Table 5.14. Comparison of performance and computational time of Algorithm 3 and Algorithm 2 under due date generation Method 1

Range Multiplier	Number of Jobs	L_{max}		d_{max}	L_{max} Difference, L_{max} (Algorithm 2) - L_{max} (Algorithm 3)	L_{max} Difference/ d_{max} (%)	Computational Time	
		Algorithm 3	Algorithm 2				Algorithm 3	Algorithm 2
0.5	20	15936.85	16278.00	753.15	341.15	45.30	0.045	0.061
0.5	50	39479.60	40350.20	781.25	870.60	111.44	0.158	0.166
0.5	100	79096.40	81639.50	791.00	2543.10	321.50	0.325	0.364
0.75	20	15585.05	15979.75	1130.60	394.70	34.91	0.045	0.061
0.75	50	39174.85	39992.85	1172.55	818.00	69.76	0.174	0.176
0.75	100	78547.55	80845.45	1187.35	2297.90	193.53	0.333	0.353
1	20	15250.65	15585.90	1507.75	335.25	22.24	0.045	0.063
1	50	38831.00	39744.00	1563.80	913.00	58.38	0.137	0.162
1	100	78343.30	80569.00	1583.55	2225.70	140.55	0.349	0.371
1.25	20	14924.20	15260.00	1884.05	335.80	17.82	0.045	0.060
1.25	50	38430.50	39305.30	1954.20	874.80	44.77	0.177	0.185
1.25	100	77804.60	80160.15	1978.85	2355.55	119.04	0.358	0.398
2	20	13881.60	14157.35	3016.10	275.75	9.14	0.049	0.061
2	50	37213.10	38254.60	3128.10	1041.50	33.29	0.143	0.162
2	100	76749.15	78840.70	3167.60	2091.55	66.03	0.340	0.363

With tighter due dates (i.e., Method 1) Algorithm 3 performed significantly better than Algorithm 2. The ratio of the difference of L_{max} to the maximum due date varied from 9.14 % to 321.50 %.

Table 5.15. Comparison of performance and computational time of Algorithm 3 and Algorithm 2 under due date generation Method 2

Range Multiplier	Number of Jobs	L_{max} Algorithm 3	L_{max} Algorithm 2	d_{max}	L_{max} Difference, L_{max} (Algorithm 2) - L_{max} (Algorithm 3)	L_{max} Difference/ d_{max} (%)	Computational Time Algorithm 3	Computational Time Algorithm 2
0.5	20	9066.00	9254.10	8328.85	188.10	2.26	0.046	0.053
0.5	50	19575.75	20865.80	21595.85	1290.05	5.97	0.157	0.151
0.5	100	36964.90	40746.45	43737.30	3781.55	8.65	0.359	0.311
0.75	20	5933.90	5547.60	12493.65	-386.30	-3.09	0.047	0.055
0.75	50	10031.10	11236.35	32393.40	1205.25	3.72	0.154	0.151
0.75	100	17239.30	20926.60	65606.30	3687.30	5.62	0.345	0.313
1	20	3027.15	2838.60	16658.25	-188.55	-1.13	0.044	0.052
1	50	3657.40	3524.80	43192.15	-132.60	-0.31	0.140	0.145
1	100	4744.15	5168.90	87475.15	424.75	0.49	0.326	0.328
1.25	20	1970.80	1727.60	20822.90	-243.20	-1.17	0.036	0.039
1.25	50	1382.50	1251.75	53989.75	-130.75	-0.24	0.127	0.113
1.25	100	1608.35	1622.65	109344.15	14.30	0.01	0.263	0.246
2	20	346.65	293.35	33317.00	-53.30	-0.16	0.021	0.024
2	50	148.90	63.90	86384.70	-85.00	-0.10	0.062	0.065
2	100	287.90	288.25	174950.80	0.35	0.00	0.157	0.130

However, with the due dates generated by Method 2, Algorithm 3 sometimes performed worse than Algorithm 2. With a large number of jobs or in the low due date ranges, Algorithm 3 performed somewhat better than Algorithm 2. The difference of L_{max} as a percentage of the maximum due date varied from 0.01 % to 8.65 % for these problems. However, with a small number of jobs in the middle and high due date ranges, Algorithm 2 performed somewhat better than Algorithm 3. The difference of L_{max} as a percentage of the maximum due date varied from –3.09 % to -0.10 %.

Average computational times were always lower for Algorithm 3 than for Algorithm 2 with the due dates generated by Method 1. With the due dates generated by Method 2, they were lower for Algorithm 3 in about half of the cases.

5.6.2. The Effect of the Initial γ and τ on the Performance of Algorithm 3

As mentioned previously, the initial γ and τ play a critical role in the performance of Algorithm 1 and its modifications including Algorithm 3. In order to investigate the effects of parameters γ and τ more systemically, the initial γ and τ were varied, and the 300 problems (i.e., 5 range multipliers × 3 number of jobs × 20 replications) were generated using Method 2 for each parameter set. γ was varied to be -6400, -3200, 0, 3200, and 6400 time units. τ was varied to be 0, 3200, 6400, 9600, and 12800 time units. Therefore, a total of 25 parameter sets was used for this experimentation. The effect of these parameters was analyzed for each combination of the range multiplier and the number of jobs. Table 5.16 summarizes the pattern of the effect and the corresponding combinations of the range multiplier and the number of jobs which are included in each category. The range multiplier seems to have a more significant relationship in these patterns than the number of jobs.

Table 5.16. The pattern of γ and τ effect

The pattern of the effect	Range multiplier / Number of jobs
Strong τ effect and weak γ effect	0.5/20, 0.5/50, 0.5/100 0.75/50, 0.75/100
Strong γ effect and weak τ effect	0.75/20 1/20, 1/50, 1/100 1.25/20, 1.25/50, 1.25/100
Strong γ effect and strong τ effect	2/20, 2/50, 2/100

Strong τ effect and weak γ effect

Figure 5.9 displays the γ and τ effect for range multipliers of 0.5 and 50 jobs as an example of a strong τ effect and weak γ effect. In this pattern, the initial τ value plays a critical role on the performance. When τ is increased, $L_{\max}$ decreases sharply and then may increase slightly before it flattens out. This pattern was especially evident in the low due date

ranges (i.e., Range multiplier of 0.5 and 0.75). When the due date range is low, there is high probability of use of the SUBSET rule. In this case, larger τ values result in more jobs being selected by the SUBSET rule than the EDD rule. This might improve the performance of Algorithm 3 but after some value (i.e., τ=3200) the τ effect becomes less significant.

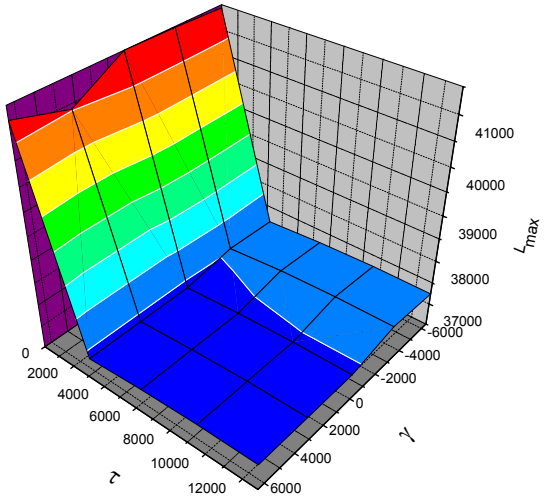
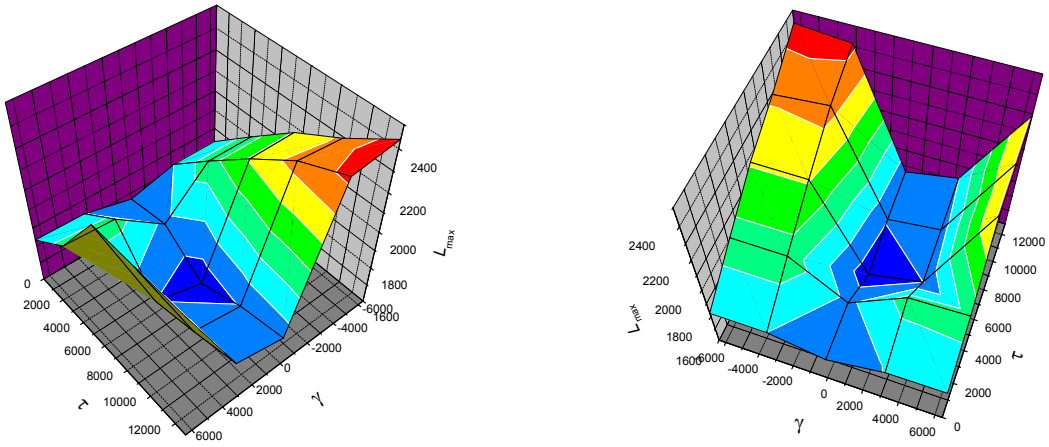


Figure 5.9. Strong τ effect and weak γ effect

Strong γ effect and weak τ effect

Figure 5.10 displays an example of a strong γ effect and weak τ effect with range multiplier of 1.25 and 100 jobs. For a better view, Figure 5.10 (a) was rotated and is shown in Figure 5.10 (b). In this pattern, the initial γ value determines whether the best performance could be attained or not. The pattern was observed in the middle due date ranges (i.e., Range multiplier of 1 and 1.25).



(a) (b)

Figure 5.10. Strong γ effect and weak τ effect

Strong γ effect and strong τ effect

Figure 5.11 displays an example of strong γ effect and a strong τ effect with a range multiplier of 2 and 100 jobs.

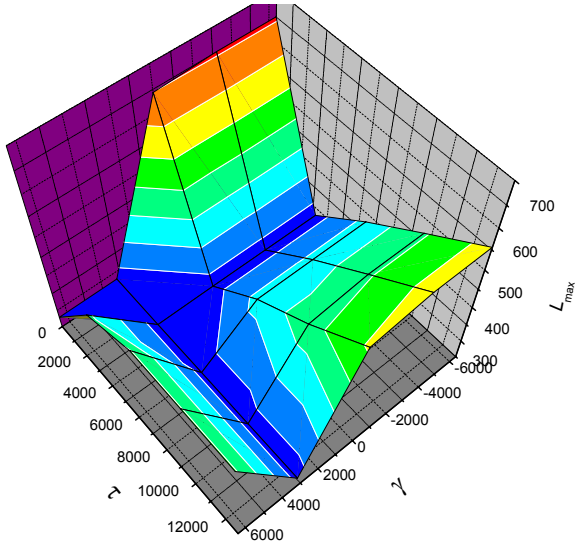


Figure 5.11. Strong γ effect and strong τ effect

In this pattern, both the parameters play a major role on the performance. When the γ value is positive, it determines the optimum performance. When the γ value is negative, the τ value plays a more critical role. This pattern was observed especially with a range multiplier of 2.

This investigation gives insight into how these parameters work in Algorithm 3. In general, in the low due date ranges, there is higher probability that most jobs are extremely tardy. In this case, the initial γ value cannot differentiate how urgent the job is. Due to the modification of M1, the initial τ value plays a more critical role to balance between selecting the SUBSET rule and the EDD rule. In the middle and high due date ranges, however, extremely tardy jobs and not urgent jobs coexist. Thus, the initial γ value tends to determine the optimum performance because it decides which rule should be used between the SUBSET rule and the SST_D rule. The observations from this investigation might help to determine the initial γ and τ values for new problems with different due date ranges so that the time and effort to find the best parameter set for optimum performance could be saved. For example, in the low due date ranges, the parameter set for the best performance could be found only by varying the initial τ value.

5.6.3. Performance Comparison to Lower Bound Approximations

The performance of Algorithm 3 was evaluated based upon the performance of Algorithm 2, and it varied in different due date ranges. In order to evaluate and understand Algorithm 3 better, the performance was compared to a computed lower bound, an absolute comparison, rather than the relative one given by comparison to other algorithms. As mentioned previously in section 5.2.2, however, developing a lower bound for the problem of

scheduling with sequence dependent setup is extremely difficult. Therefore, in the lower bound calculation, the operation with sequence dependent setup was substituted by an operation with sequence independent setup time. The total amount of time used for setup was evenly distributed to all the jobs. Since the amount of time used for each setup was not known, three scenarios in which every job has setup time of zero, quarter of a setup time, and half of a setup time were considered. Then a *LB* was computed in the same manner as for the original VF. While the *LB* with setup time of 0 is an actual *LB* on the problem with sequence dependent setups, it may be a very weak *LB*. Therefore the *LB* with a quarter and a half of a setup time will be used as a *LB* approximation to the *LB* on the problem with sequence dependent setups. To test the performance of Algorithm 3 against these, the due date range was varied between 0 to 160000, where 200 time units represented 1 hour.

In Figure 5.12, 5.13, and 5.14, the difference between the attained $L_{\max}$ and *LB* is displayed as a function of the due date range with 20, 50 and 100 jobs, respectively. These results are the average of 20 random instances (replications) generated. Table 5.17, 5.18, and 5.19 show the relationship between the range multiplier and corresponding due date range with 20, 50 and 100 jobs, respectively.

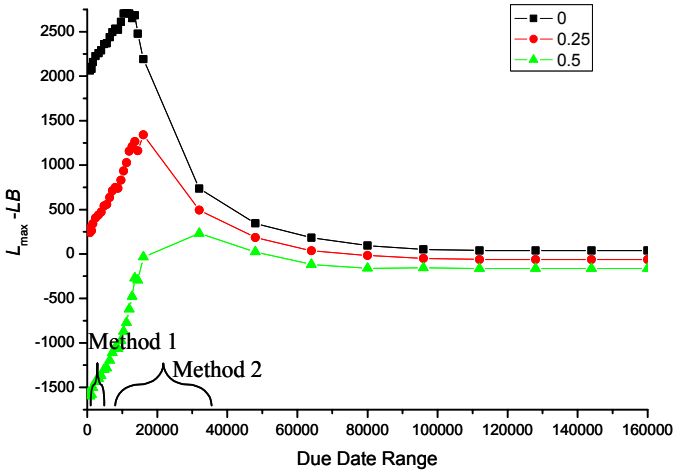


Figure 5.12. Performance of Algorithm 3 to *LB* approximation with 20 jobs

Generation Method	Method 1					Method 2				
Range Multiplier	0.5	0.75	1	1.25	2	0.5	0.75	1	1.25	2
Due Date Range	799.5	1199	1599	1999	3198	8830	13245	17660	22075	35320

Table 5.17. Relationship between range multiplier and due date range for 20 jobs

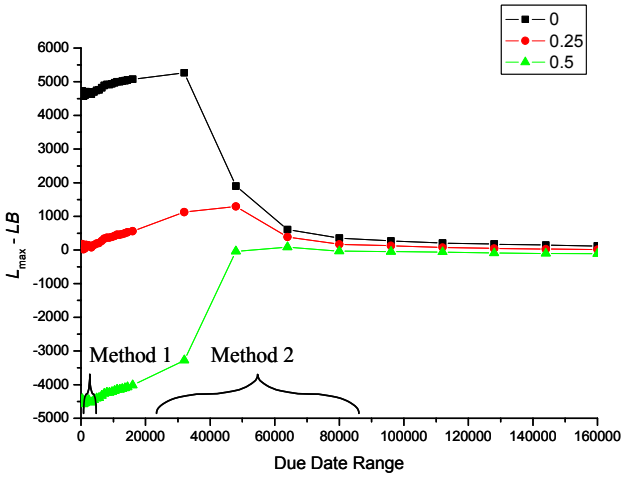


Figure 5.13. Performance of Algorithm 3 to *LB* approximation with 50 jobs

Generation Method	Method 1					Method 2				
Range Multiplier	0.5	0.75	1	1.25	2	0.5	0.75	1	1.25	2
Due Date Range	799.5	1199	1599	1999	3198	22075	33113	44150	55188	88300

Table 5.18. Relationship between range multiplier and due date range for 50 jobs

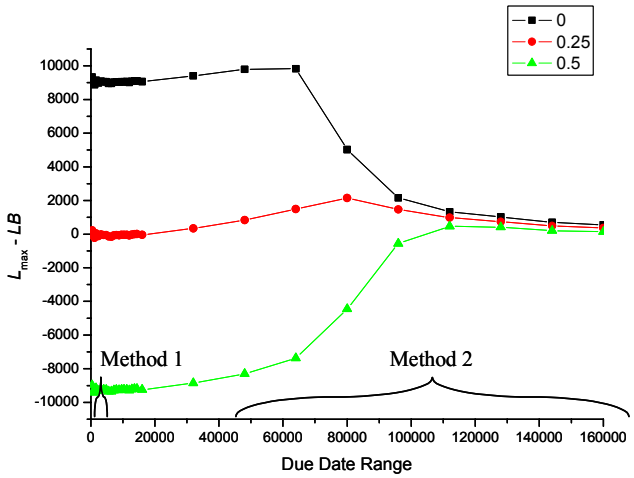


Figure 5.14. Performance of Algorithm 3 to *LB* approximation with 100 jobs

Generation Method	Method 1					Method 2				
Range Multiplier	0.5	0.75	1	1.25	2	0.5	0.75	1	1.25	2
Due Date Range	799.5	1199	1599	1999	3198	44150	66225	88300	110375	176600

Table 5.19. Relationship between range multiplier and due date range for 100 jobs

Regardless of the number of jobs and the amount of setup time, the difference between L_{max} and *LB* initially became larger as the due date range increased. Then, after a certain point over the due date range, the difference flattened out. This point over due date range can be estimated by the following equation:

(Average processing time of dyeing operation + setup time) × Number of Jobs

which is the due date range with range multiplier of 1 under Method 2. For example, with a quarter of a setup time and 100 jobs, the difference has its largest value at the due date range of 80,000 (Figure 5.16) and the corresponding computed due date range is (700+366×0.25) ×100 = 79,150.

With 20 jobs, the maximum due date range was 3,198 and 35,320 on problems generated by Method 1 and Method 2, respectively. With 50 jobs, the maximum due date range was 3,198 and 88,300 on problems generated by Method 1 and Method 2, respectively. With 100 jobs, the maximum due date range was 3,198 and 176,600 on problems generated by Method 1 and Method 2, respectively. The two brackets in Figures 5.12, 5.13 and 5.14 show the due date ranges generated by each method with 20, 50, and 100 jobs, respectively. The region where problems were generated by Method 1 covers a very small part of the due date range on the graphs of the performance to *LB*. The region where problems were generated by Method 2 falls in the due date ranges both before and when the difference flattens out on the graphs of the performance to *LB*. Also the region covered fairly high due date ranges. The maximum due date range increased as the number of jobs is increased.

CHAPTER 6. CONCLUSIONS AND FUTURE RESEARCH

6.1. Conclusions

Textile coloration firms have tried to stay competitive in rapidly changing business conditions and intense global competition due to the lower labor costs of developing countries. Since textile coloration firms must be more flexible and more responsive to customer needs for their long-term survival, meeting customer due dates and reducing lead time have become important. Batchwise fabric dyeing operations are one of the areas where this performance can be improved since they have significant sequence dependent setup times. In this dissertation, performance optimization of the fiber-textile-apparel-retail chain, which includes batchwise fabric dyeing operations, was pursued.

First, experiments of scheduling multiple entities within the supply chain simultaneously were conducted. The performance of the VF was tested in multi-factory, rolling horizon settings including *Two Factories in Series*, *Three Factories in Series*, *Two Factories Feeding One* and *One Factory Feeding Two*. Considering the total horizon length of 100 days and the number of jobs processed, the VF performed well in all multi-factory supply chain environments. Also the VF performed robustly when parameters were varied such as the number of jobs released, truck capacity, total horizon length, and warm-up period.

Then, specific characteristics of dyeing operations within the textile supply chain were investigated in order to improve their performance. Additional complications in applying Taner's matrix were addressed and the setup structure was expanded to include four indices; fabric/dye type, hue, lightness and chroma, by taking the theoretical approach of color physics. In order to analyze the interdependency of the hue, lightness, and chroma of

colors dyed within the same fabric/dye type for determining setups, the problem was first approached graphically in three dimensional color space. Then, matrices were developed to capture this interdependency. The differences between the assumed cylindrical color space and the actual color space were addressed.

Finally, Taner's sequence dependent scheduling algorithm according was refined according to the dyeing matrix that was developed. Algorithm 1, including a new SST rule for dyeing operations (SST_D), was proposed. A new due date generation method, Method 2, was developed to test higher due date ranges than those generated by Method 1, Taner's original due date generation method. Algorithm 1, Algorithm 2 (the modified LAO (β) rule), and Algorithm 1 modifications including Algorithm 3 were implemented in the VF, and a variety of one machine flowshop scenarios were generated and tested with each algorithm. In comparison to Algorithm 2, Algorithm 3 performed well with the tight due dates or with a large number of jobs but not with other parameters. Sensitivity analysis of the initial γ and τ of Algorithm 3 showed that its performance is highly dependent on these values. A lower bound approximation was calculated as an alternate way of gauging the performance of Algorithm 3.

6.2. Future Research

In defining the dyeing setup matrices, certain regions in the cylindrical color space model used in this dissertation are not present in the object color system. Refining this cylindrical space model based on the object-color system would enable us to generate more realistic problems since the jobs would have index coordinates which represent true colors. Moreover, the matrices should be verified by collecting data from industry. Time needed to

perform the machine and operator activities for the setup process needs to be studied and recorded.

The algorithm proposed for scheduling batchwise fabric dyeing operations should be refined in order to improve its performance in all due date ranges. Taner's original algorithm for sequence dependent setup times was based on a queuing model (Markov Decision Process). The scheduling policy based on the experimentation with the model might not be optimal for the general problem of sequencing operations with sequence dependent setups even with Taner's specific setup matrix since some of the parameters might have not been varied sufficiently, including due date assignment, size of the queue, and specified conditions in which a particular scheduling rule is chosen. Thus, it is not certain that the system would behave in the same manner when other parameters are changed. Additional experimentation could be performed varying the parameters. A scheduling policy developed based on the results might be different than Taner's original characterization of the optimal policy. Moreover, for dyeing problems, the actual dyeing matrix developed could be used in a Markov Decision Process.

The algorithm developed is applicable when jobs of the same fabric/dye type are sequenced in one dyeing machine. Thus, when dyeing plants have dedicated machines for each fabric/dye type, the proposed algorithm might adequately schedule the jobs in various due date ranges. The one machine dyeing problem should be extended to include the fabric/dye type consideration. Then, research could be undertaken to develop an algorithm for the parallel dyeing machine problem. Similar to the one machine dyeing problem, the fabric/dye type consideration could be excluded as an initial step when developing the

solution. Then, an algorithm which enables scheduling jobs of various fabric/dye type, hue, lightness and chroma on dyeing machines in parallel could be established.

In order to test performance of the proposed algorithm, a transient implementation of the Virtual Factory was used. To better test its performance, experiments should be performed in a rolling horizon scenario. After testing one-factory, rolling horizon experiments with batchwise fabric dyeing operations in the VF, research could be extended to develop multi-factory problems. In industry, yarn and fabric formation, dyeing and finishing, and cutting and sewing operations may take place in different factories. Various supply chain configurations could be explored with batchwise fabric dyeing operations. These include a *Two Factories in Series* scenario in which fabric is batchwise dyed in one factory and cut and sewn in another and a *Three Factories in Series* scenario where Factory 1, 2, and 3 perform yarn and fabric formation, dyeing and finishing, and cutting and sewing operations, respectively.

REFERENCES

1. Allahverdi, A., Gupta, J., and Aldowaisan, T. (1999) "A review of scheduling research involving setup considerations", *Omega-International Journal of Management Science,* 27, 219-239.
2. Baker, K. R. (1984) "Sequencing rules and due-date assignments in a job shop", *Management Science*, 30 (9), 1093-1104.
3. Baker, K. R. (1999) "Heuristic procedures for scheduling job families with setups and due dates", *Naval Research Logistics*, 46, 978-991.
4. Bearpark, I., Marriott, F. W. and Park, J. (1986) "Practical introduction to the dyeing and finishing of wool fabric", *JSDC*, The Eastern Press Ltd.
5. Burdett, B. C. and King, A. J. (1999) "The dyehouse into the 21st century", *Rev. Prog. Coloration* 29, 29-36.
6. Burnham, R. W., Hanes, R. M., and Bartleson, C. J. (1963) Color: A guide to basic facts and concepts, John Wiley & Sons, Inc, New York.
7. Clarke, Geoffrey (1982) A practical introduction to fibre and tow coloration, Society of Dyers and Colourists on behalf of the Dyers' Company Publications Trust.
8. Cooper, W. D. and Saydam, C. (1998) "Maximize dyeing ROI via optimal port scheduling", *Textile World*, 148(10), 85-90.
9. Das, S. R., Gupta, J. N. D., and Khumawala B. M.(1995) "A saving index heuristic algorithm for flowshop scheduling with sequence dependent setup times", *Journal of the Operational Research Society*, 46 (11), 1365-1373.

10. Demirkol, E., E. Mehta, and R. Uzsoy. (1998) "Benchmarks for shop scheduling problems", *European Journal of Operational Research*, 109, 137-141.
11. EPA/625/R-96/004 (1996) "Manual: Best management practices for pollution prevention in the textile industry", U.S. Environmental Protection Agency, Office of Research and Development, National Risk Management Research Laboratory Center for Environmental Research Information, Cincinnati, Ohio.
12. Fralix, M. T. (2000) "Fabric printing in a totally digital supply chain", A doctoral dissertation submitted to North Carolina State University, Textile Technology and Management.
13. 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 times", *International Journal of Production Economics*, 43, 79-89.
14. Gere, W. S., Jr (1966) in Blackstone Jr., J. H., Phillips, D. T., and Hogg, G. L. (1982) "A state-of-the-art survey of dispatching rules for manufacturing job shop operations", *Int. J. Prod. Res*., 20 (1), 27-45.
15. Glassey, C. R. and Weng, W. W. (1991) "Dynamic batching heuristic for simultaneous processing", *IEEE Transactions on Semiconductor Manufacturing*, 4 (2), 77-82.
16. Gould, L. S. (2002) "Introducing APS: Getting production in lock step with customer demand", www.autofieldguide.com/articles/article_print.cfm, (04/01/2002).
17. Handfield, R.B. and E.L. Nichols, Jr. (2002) Supply chain redesign: Transforming supply chain into integrated value systems, Prentice Hall.
18. Hodgson, T. J., Cormier, D., Weintraub, A. J., and Zozom, A. (1998) "Note. Satisfying due date in large job shop", *Management Science*, 44 (10), 1442-1446.

19. Hodgson, T. J., King, R. E., Thoney, K.A., Stanislaw, N., Weintraub, A. J., and Zozom, A. (2000) "On Satisfying due date in large job shop: Idle time insertion", *IIE Transactions*, 32 (7), 177-180.
20. Hoffmann, F. (1998) "Computer-Aided optimization of exhaust dyeing procedures", *Textile Chemist and Colorist*, 30 (10), 19-23.
21. Holme, I. (2001) "Quest for consistency in batch dyeing", *International Dyer,* November, 9-10.
22. Imperial Chemical Industries (ICI) Limited Dyestuffs Division. (1964) The dyeing of polyester fibres, third edition, Raithby, Lawrence & Co. Ltd., De Montfort Press, Leicester.
23. Internet source (a). (2003) "Cleaning agent for dyeing machines; NEOPOLAN KC-10", http://textileinfor.com/en/chemicals/takamatsu/seikou/page01.html, (11/04/2003).
24. Internet source (b). (2003) "Auxiliaries for a system of correction of faulty cloth", http://textileinfo.com/en/chemicals/nipponkayaku/furyou/page03-01.html, (11/04/2003).
25. Kelton, W. D., Sadowski, R. P., and Sadowski, D. A. (2002) Simulation with Arena, second edition, McGraw Hill.
26. Kim, S. C. and Bobrowski, P.M. (1994) "Impact of sequence-dependent setup times on job shop scheduling performance", *International Journal of Production Research*, 32 (7), 1503-1520
27. King, R. E., Hodgson, T. J., Little, T. J., and Thoney, K. A. (2002) "Emerging apparel supply chain configurations", http://www.ntcresearch.org.

28. Koksal, G. and Smith Jr., W. A. (1990) “System analysis for dyehouse quality control”, IE Technical Report 90-3, Industrial Engineering Department, NC State University, April.

29. Koksal, G., Smith Jr., W. A. and Smith, B. (1992) “A system analysis of textile operations: A modern approach for meeting customer requirements”, *Textile Chemist and Colorist*, 24 (10), 30-35.

30. Koksal, G., Smith Jr.,W. A., Fathi, Y., Lu, J., and MaGregor, R. (1998) “ A case study in off-line quality control: Characterization and optimization of batch dyeing process design”, *Int. J. Technology Management*, 16 (4/5/6), 358-382.

31. Krajewski, L.J., King, B.E., Ritzman, L. P., and Wong, D. S. (1987) “Kanban, MRP, and shaping the manufacturing environment”, *Management Science*, 33 (1), 39-57.

32. Madaras, G. W., Parish, G. J. and Shore, J. (1993) “Batchwise dyeing of woven cellulosic fabrics”, *JSDC*, Staples printers Rochester Ltd.

33. Maldonado, F., Ciurlizza, A., Radillo, R. and Ponce, E. (2000) “Optimisation of the colour sequence in the dyeing process: Industrial applications”, *JSDC,* 116 (November), 359-362.

34. McDonald, Roderick. (1987) Colour physics for industry, Society of Dyers and Colourists on behalf of the Dyers’ Company Publications Trust.

35. Mock, G. N. (1997) “ Fundamentals of dyeing and printing”, Lecture notebook, College of Textiles, NCSU.

36. Morales, L., Maldonado, F., Radillo, R. and Ciurlizza, A. (1996) “Optimisation of the colour sequence in the process of fabric dyeing”, *JSDC,* 112 (December), 361-363.

37. Monma, C. L. and Potts, C. N. (1989) "On the complexity of scheduling with batch setup times", *Operations Research*, 37 (5), 798-804.

38. Moor, C. (1997) "Dyehouse management for the 21st century", *International Dyer*, 182 (8), 32-34.

39. Morton, T.E. and Lawrence, S.R. (1986) "Patriarch: Hierarchical production scheduling", *National Bureau of Standards Special Publication*, 724, 87-97.

40. Nanda, B. (1998) "Production management in a dyehouse", *Colourage* Texindia fair '98 special/ 6th-8th, November, 41-48.

41. Nara Lace, Co. Ltd (2003) "Dye manufacturer chooses Asprova for high speed scheduling process and user-friendly GUI environment", http://www.asprova.com/en/kaizen/case-nl/nara-lace.html (02/02/2003).

42. O'Grady, P. J. and Harrison, C. (1988) "Search based job scheduling and sequencing with setup times", *OMEGA*, 16, 547-552.

43. Ovacik,I. M. and Uzsoy, Reha (1994a) "Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times", *Int. J. Prod. Res.*, 32 (6), 1243-1263.

44. Ovacik, I. M. and Uzsoy, R. (1994b) "Exploiting shop floor status information to schedule complex job shops", *Journal of Manufacturing Systems*, 13 (2), 73-84.

45. Pajgrt, O. and Reichstadter, B. (1979) Processing of polyester fibres, Elsevier Scientific Publishing Company.

46. Panwalkar, S. S., Dudek, R. A., and Smith, M. L. (1973) "Sequencing research and the industrial scheduling problem", Symposium on the Theory of Scheduling and its Application, 29-38.

47. Patterson, M. C. (1993) "Analysis of setup time at constraint resources", *International Journal of Production Research*, 31 (4), 845-849.

48. Simchi-Levi, D., P.Kaminsky, and E. Simchi-Levi. (2000) Designing and managing the supply chain: Concepts, strategies, and case studies, McGraw-Hill.

49. Srikar, B. N., and Ghosh, S. (1986) "A MILP model for the n-jobs, M-stage flowshop, with sequence dependent setup times", *International Journal of Production Research*, 24 (6), 1459-1472.

50. Tan, K., Narasimhan, R., Rubin, P. A., and Ragatz, G. L. (2000) "A comparion of four methods for minimizing total tardiness on a single processor with sequence dependent setup times", *OMEGA*, 28, 313-326.

51. Taner, M. R. (2001) "Scheduling with sequence dependent setup times", A dissertation submitted to the Graduate faculty of North Carolina State University for the degree of Doctor of Philosophy Industrial Engineering.

52. Taylor, D. and D. Brunt. (2001) Manufacturing operations and supply chain management: The LEAN approach, Thomson Learning.

53. Thoney, K.A (2000) "Simultaneous plant and supply chain scheduling", A dissertation submitted to the Graduate faculty of North Carolina State University for the degree of Doctor of Philosophy Industrial Engineering.

54. Thoney, K.A., T.J. Hodgson, R.E. King, M.R. Taner, and A.D. Wilson. (2002a) "Satisfying due-dates in large multi-factory supply chains", *IIE Transactions*, 34, 803-811.

55. Thoney, K.A., J.A. Joines, P. Manninagarajan, and T.J. Hodgson. (2002b) "Rolling horizon scheduling in large job shops", *Proceedings of the 2002 Winter Simulation Conference*.

56. Unknown. (1997) "Gaston County: SuperTex +SQL Dyehouse management system", *Textile World*, 147, 76-77.

57. Vepsalainen, A.P.J. and T.E. Morton. (1988) "Improving local priority rules with global lead-time estimates: A simulation study", *Journal of Manufacturing and Operations Management*, 1, 102-118.

58. Wortman, D. B. (1992) "Managing capacity: getting the most from your firm's assets", *Industrial Engineering*, 24 (2), 47-49.

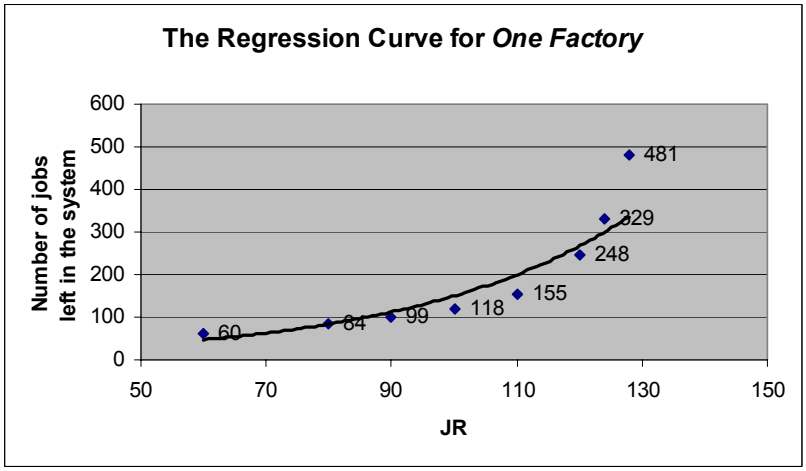
59. Yang, J, and Deane, R.H. (1993) "Setup reduction and competitive advantage in a closed manufacturing cell", *European Journal of Operational Research*, 69, 413-423.

60. Zhou, C. and Egbulu, P. J. (1989) "Scheduling in a manufacturing shop with sequence-dependent setups", *Robotics and Computer-Integrated Manufacturing*, 5, 73-81.

APPENDIX A. REGRESSION CURVE USED FOR EXPERIMENTS

A. 1. Regression Curve from Excel (Office 2000)

Figure A.1 shows the regression curve for the empirical N_s value from Excel 2000. Based on this curve, *JR* of 102, 115, and 122 jobs (which are 80%, 90% and 95% workload, respectively) have empirical N_s value of 159, 232, and 284 jobs, respectively. Figure A.2, A.3, and A.4 show the results with *JR* of 80%, 90% and 95% workload, respectively. The steady state for rolling horizon scenario was achieved within 10 days in all cases. The system stabilized at the value of 124, 183, and 274, respectively. The values from the regression curve are somewhat higher than the actual values at which the system stabilizes since the curve does not fit well when *JR* values are larger than 100 jobs.



$$y = 8.3586e^{0.0289x}$$
$$R^2 = 0.9068$$

Figure A.1. Regression Curve from Excel 2000

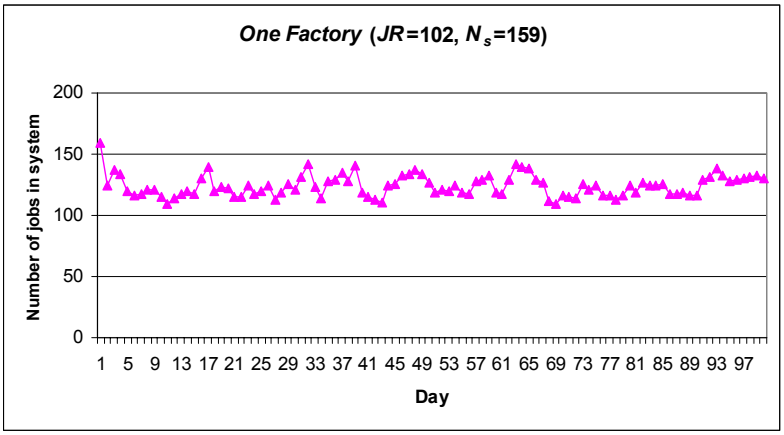


Figure A.2. The number of jobs in the system with *JR* of 102 (80 % workload) and N_s of 159

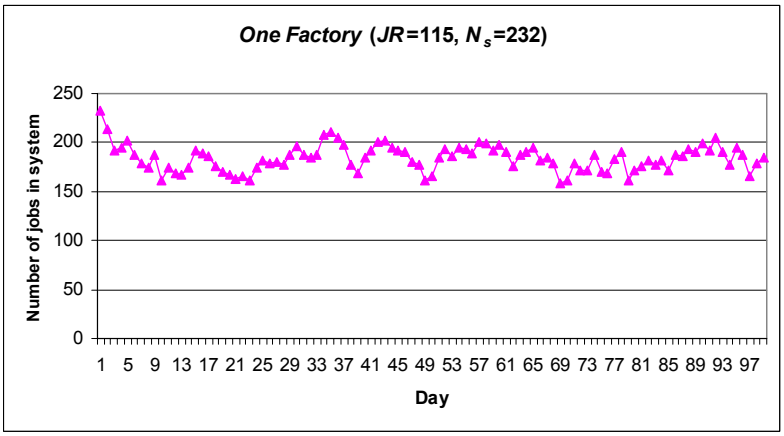


Figure A.3. The number of jobs in the system with *JR* of 115 (90 % workload) and N_s of 232

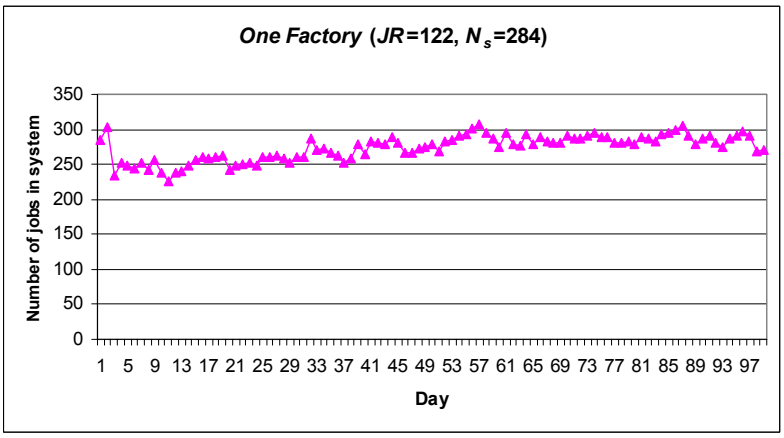
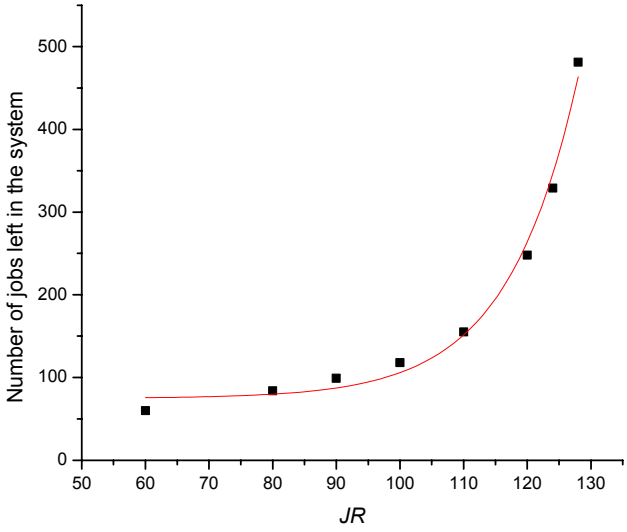


Figure A.4. The number of jobs in the system with *JR* of 122 (95 % workload) and N_s of 284

A. 2. Regression Curve from Origin 7.0

Figure A.5 shows the regression curve for the empirical N_s value from Origin 7.0. Based on this curve, *JR* of 102, 115, and 122 jobs (which are 80%, 90% and 95% workload, respectively) have empirical N_s value of 112, 195, and 301 jobs, respectively. Figure A.6, A.7, and A.8 show the results with *JR* of 80%, 90% and 95% workload, respectively. The steady state for the rolling horizon scenario was achieved within 5 days. Similar to the results in A.1, the system also stabilized at the value of 124, 183, and 274, respectively.



$$y = 0.00376e^{x/11.08596} + 74.81516$$
$$R^2 = 0.99073$$

Figure A.5. Regression curve from Origin 7.0

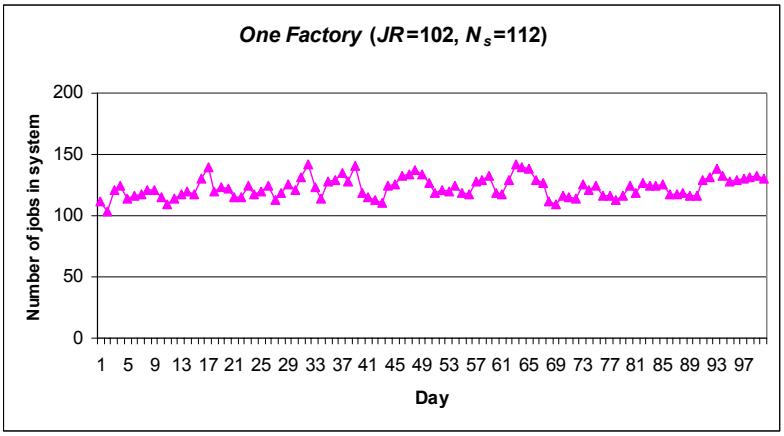


Figure A.6. The number of jobs in the system with *JR* of 102 (80 % workload) and N_s of 112

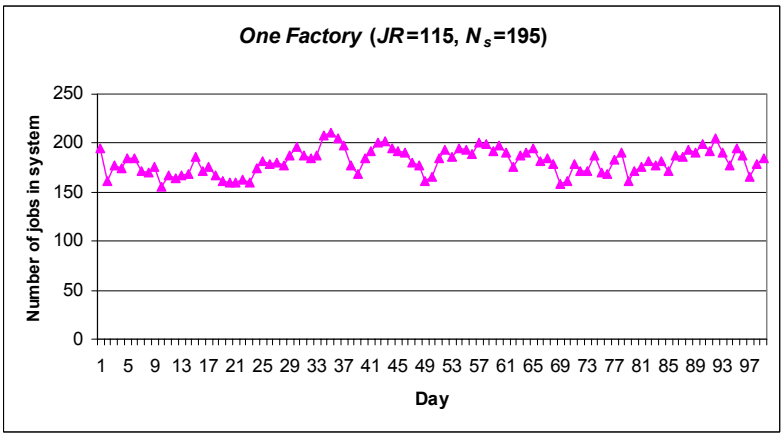


Figure A.7. The number of jobs in the system with *JR* of 115 (90 % workload) and N_s of 195

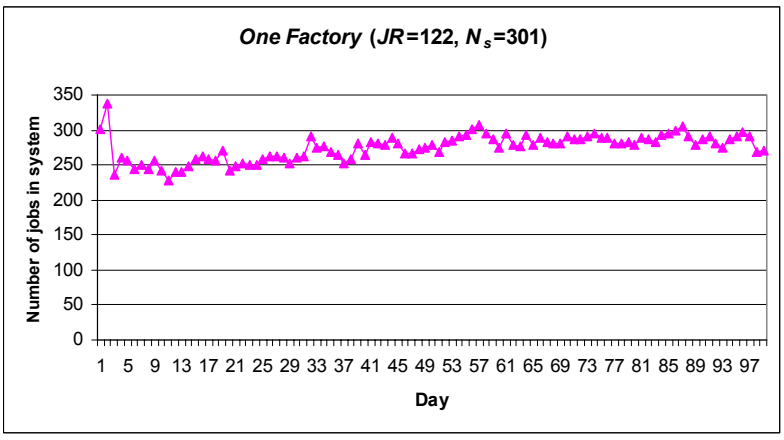


Figure A.8. The number of jobs in the system with *JR* of 122 (95 % workload) and N_s of 301

A. 3. The Actual Value at which the System Stabilized

From the results of A.1 and A.2, *JR* of 102, 115, and 122 jobs (which are 80%, 90% and 95% workload, respectively) have the actual values at which the system stabilizes of 124, 183, and 274, respectively. Figure A.9, A.10, and A.11 show the results with *JR* of 80%, 90% and 95% workload, respectively. The steady state for rolling horizon scenario was achieved within 5 days. The values at which the system stabilized are the same as 124, 183, and 274, respectively.

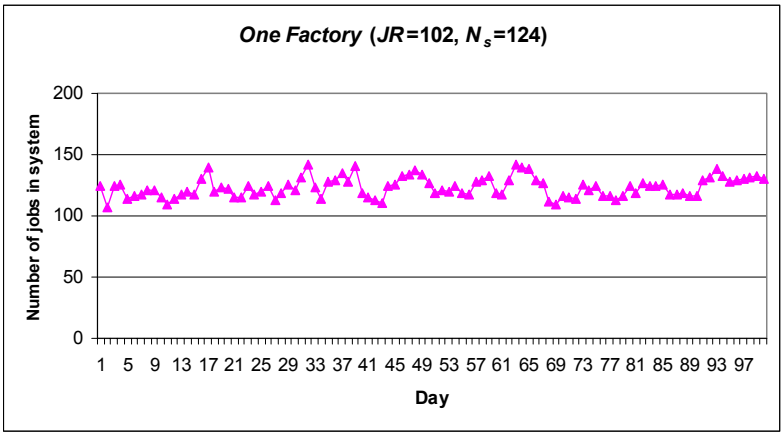


Figure A.9. The number of jobs in the system with *JR* of 102 (80 % workload) and N_s of 124

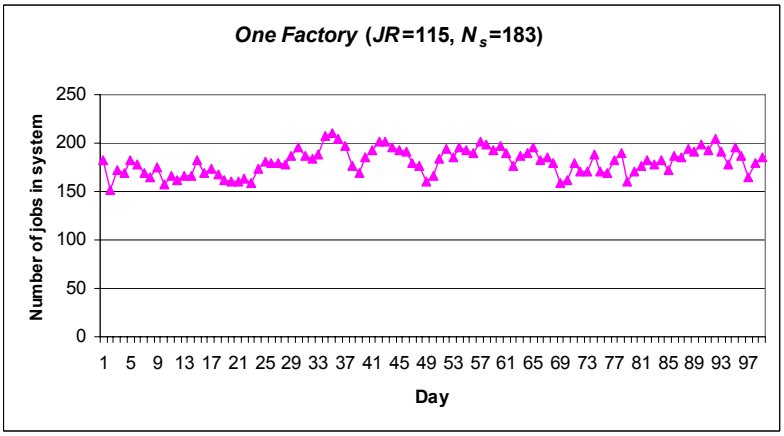


Figure A.10. The number of jobs in the system with *JR* of 115 (90 % workload) and N_s of 183

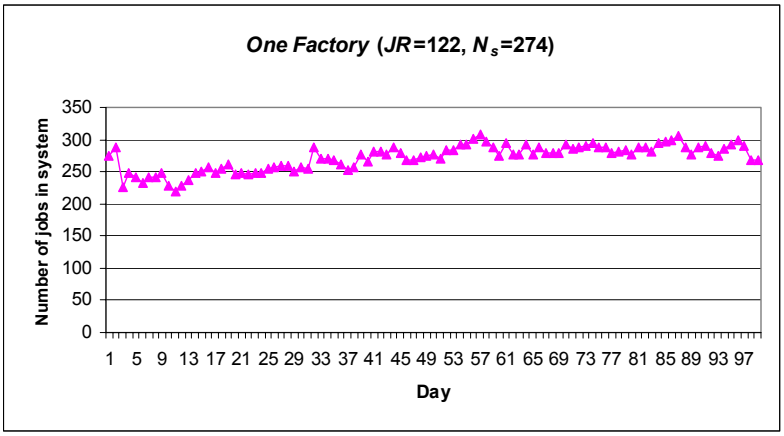


Figure A.11. The number of jobs in the system with JR of 122 (95 % workload) and N_s of 274

A.4. VF Performance to Meeting Due Dates

Results showed that steady state for the rolling horizon scenario was achieved within 10 days in all cases, regardless of which empirical N_s values were used. Furthermore, for the same *JR* value and different empirical N_s values, the system stabilized to the same number of jobs. Since a warm-up period of at least 10 days was used in all rolling horizon experimentation in this dissertation, we conjectured that the performance of the VF to meeting due dates would not be significantly affected if one of these methods was used to determine the empirical N_s over the other. To test whether this conjecture was true, the *One Factory* problem was rerun where the initial values used were those at which the system actually stabilized (A.3). These results can be seen in Figure A.12. There is not much difference between this graph and that in which the empirical N_s was computed from the regression curve made in Excel (Figure 3.5). Thus, it seems that our conjecture is correct, and the results in Chapter 3 would not be significantly changed by instead using the actual value at which the systems stabilized.

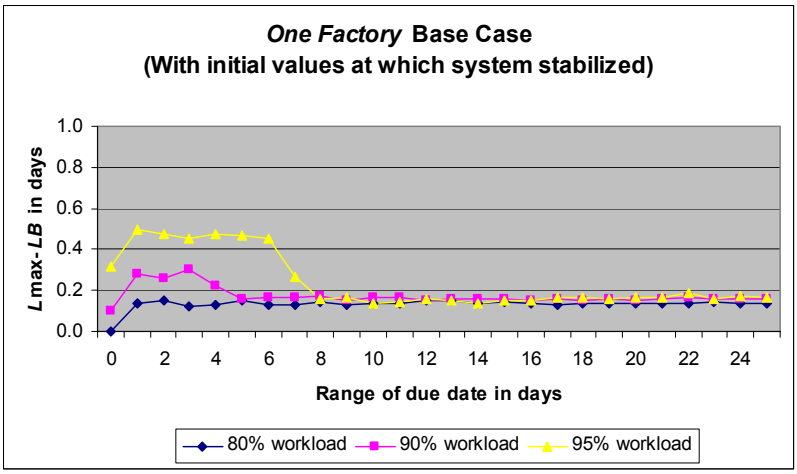


Figure A.12. Performance of the base case of *One Factory* using initial values at which system stabilized

APPENDIX B. DYEING PROFILE OF EACH FABRIC AND CORRESPONDING DYE GROUP

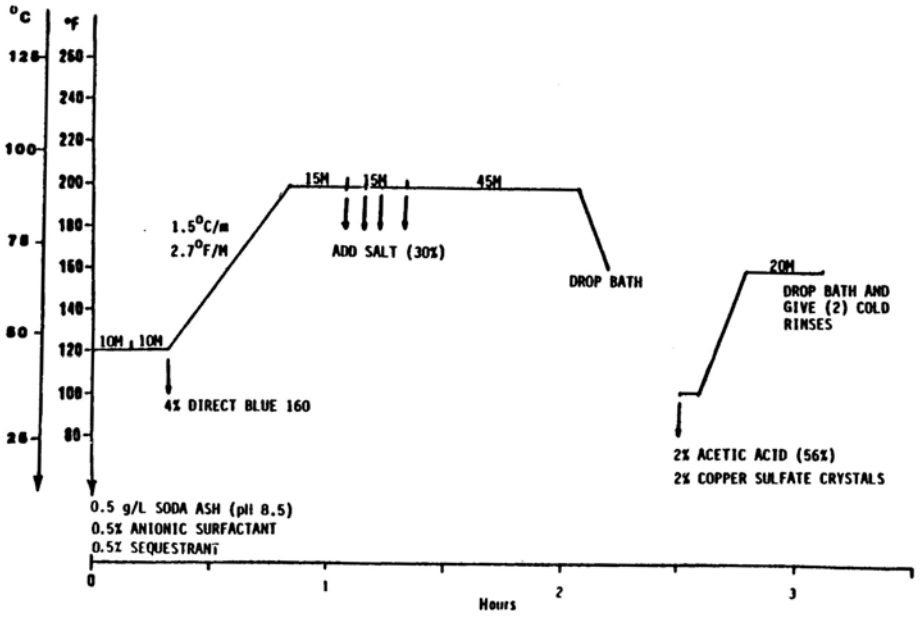


Figure B.1. Dyeing bleached mercerized cotton with direct dyes [Kulkarni et al.,1986]

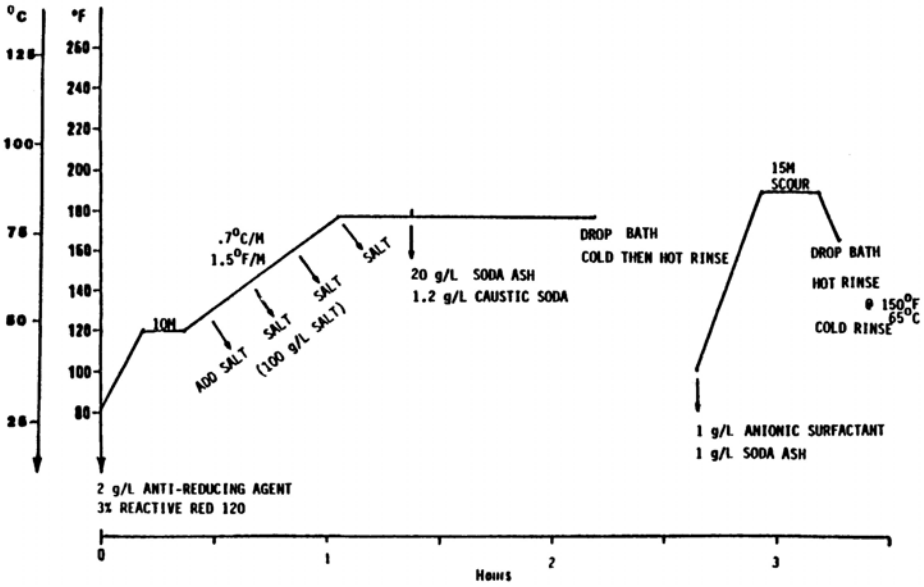


Figure B.2. Dyeing bleached mercerized cotton with reactive dyes [Kulkarni et al.,1986]

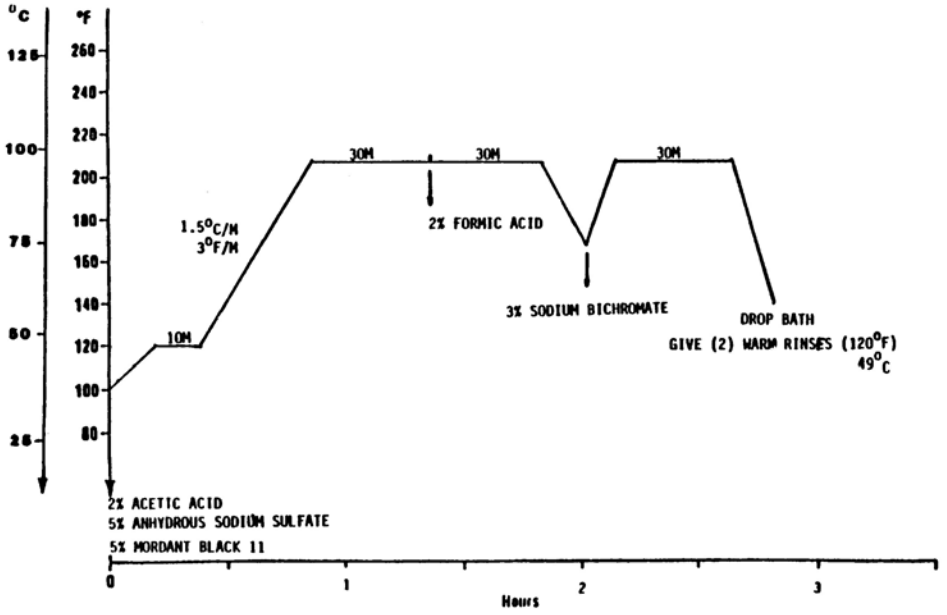


Figure B.3. Dyeing wool with acid dyes [Kulkarni et al.,1986]

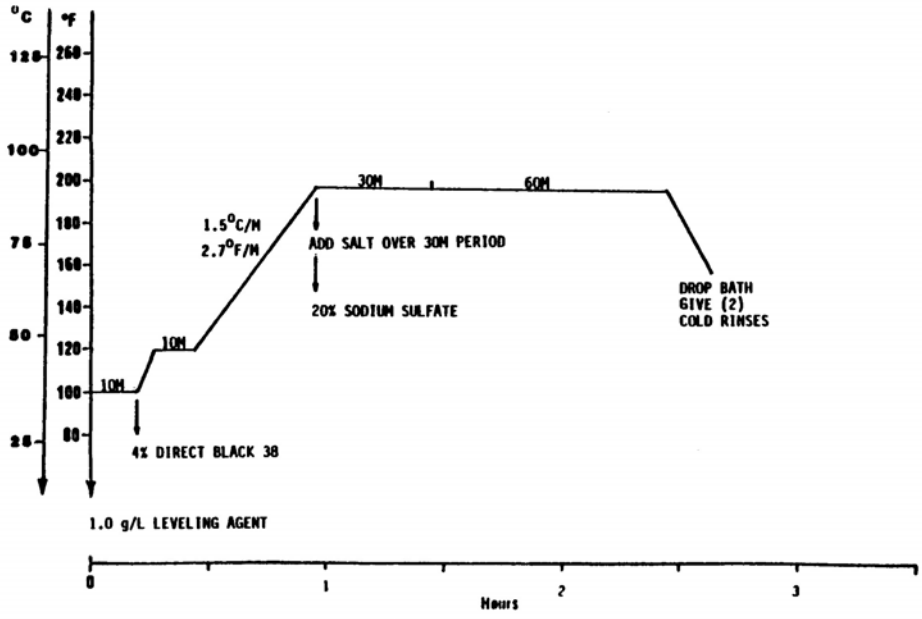


Figure B.4. Dyeing rayon with direct dyes [Kulkarni et al.,1986]

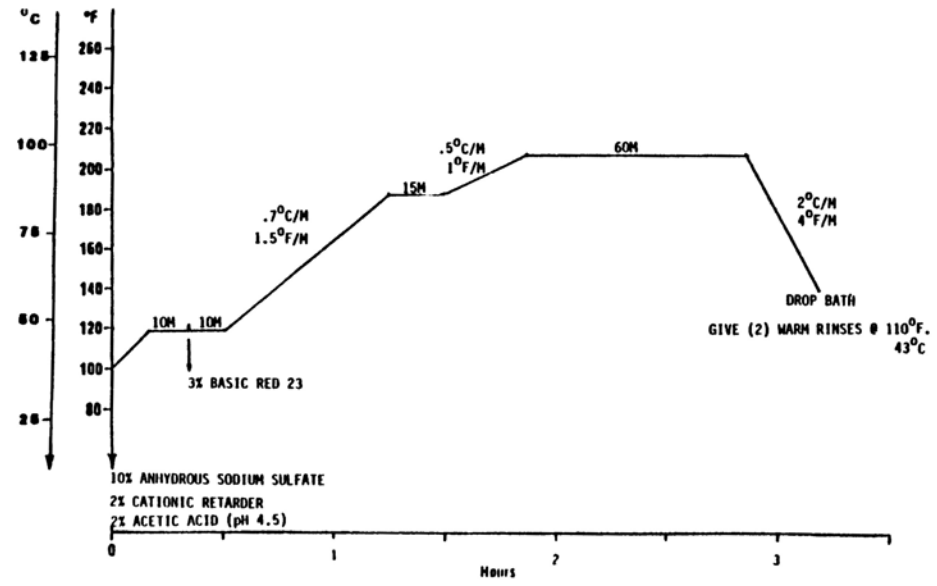


Figure B.5. Dyeing acrylic with basic dyes [Kulkarni et al.,1986]

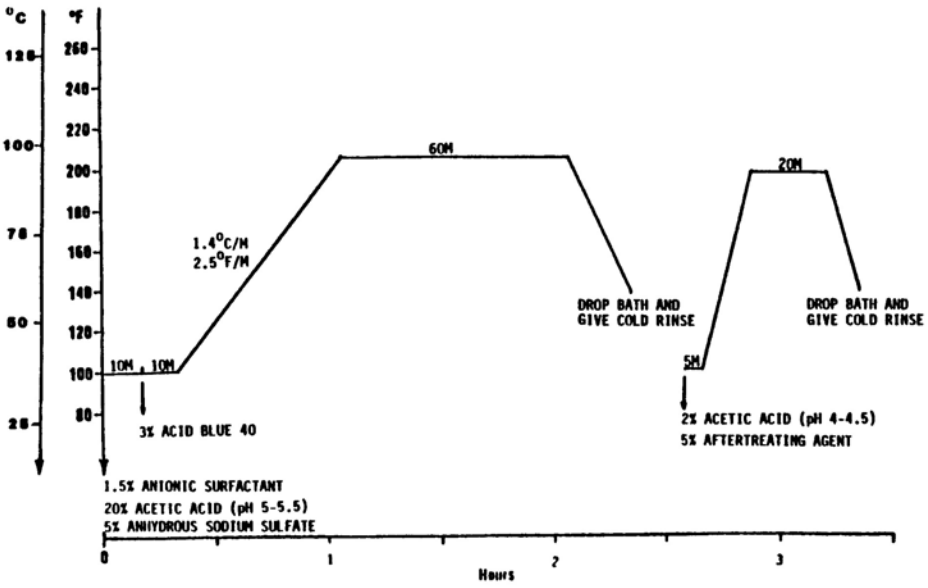


Figure B.6. Dyeing polyamide with acid dyes [Kulkarni et al.,1986]

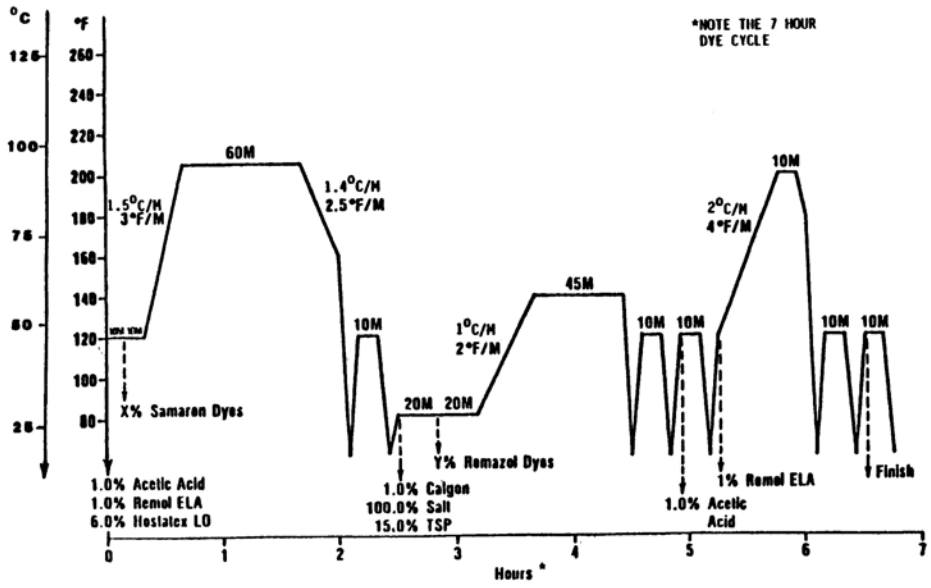


Figure B.7. Dyeing polyester/cotton blend with disperse/reactive dyes [Kulkarni et al.,1986]